HIGH PERFORMANCE CLUSTER AND
GRID COMPUTING SOLUTIONS
FOR SCIENCE

By

UMESHKUMAR KESWANI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2008

ACKNOWLEDGEMENTS

I would like to thank my supervising faculty, David Levine, Department of Computer Science and Engineering for his constant encouragement and being an excellent guide in my research work.

I would like to thank Cédric Feschotte, Department of Biology for having provided direction in my thesis work and for having a great deal of patience in explaining the intricacies of the biology involved.

I would like to thank Gergely Záruba, Department of Computer Science and Engineering for serving on my committee.

I would like to thank Kaushik De and Jaehoon Yu for providing me with an opportunity to work at the Brookhaven National Laboratory.

I would like to thank my family and friends who provided constant support and helped me with my other obligations.

July 2, 2008

ABSTRACT

HIGH PERFORMANCE CLUSTER AND GRID COMPUTING SOLUTIONS FOR SCIENCE

Umeshkumar Keswani, M.S.

The University of Texas at Arlington, 2008

Supervising Faculty: David Levine

Computing for science has very peculiar requirements. There is an enormous amount of data to analyze to get useful information. This requires greater computing power, huge data storage and high-capacity network. High Performance Computing (HPC) is the perfect solution that satisfies these requirements. Here, we look at two high performance implementations that provide useful solutions to computing problems in science, making possible scientific experiments and discoveries that otherwise would not have been possible.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Physical sciences are about understanding nature. Typically, this process involves gathering data and analyzing it to get some useful scientific information. Scientists are always looking for tools that help them achieve this goal. Computer science, which is an applied science, has provided and continues to provide many such tools. It has helped automate many a manual process. Reducing the time spent in gathering and analyzing data from months to weeks and in some cases to days. The collaboration between physical sciences and computer science has been beneficial for computer science as well. The hunger of scientists for new discoveries is driving computer science professionals to come up with better tools that can assist in this endeavor.

Improvements in technology have made possible scientific experiments that were earlier unthinkable. These experiments have generated an enormous amount of data at a shockingly brisk pace. Without the help of faster and better computing tools analyzing this data might take scientists several lifetimes. Computing tools can be made faster by employing more computing power. This additional computing power can be garnered in multiple ways. One of the methods is to put more transistors on a single chip making processors execute applications faster. However, this approach alone is not sustainable since it leads to more heat dissipation hence, more power consumption.

A better approach is to put multiple processors on multiple chips inside the same machine. This approach requires that the task performed by a computing tool to be divided into multiple sub-tasks that can be executed simultaneously on separate processors. However, there is a practical limit to the number of processors that can be placed in a single machine. All the processors in the machine have to contend for resources such as access to the memory bus. Hence, after a certain limit adding more processor does not enhance the computer performance. To overcome this limitation, in addition to employing multiple processors in a single machine, multiple machines can be employed. This approach is known as High Performance Computing (HPC).

1.1 Cluster Computing

A cluster is group of similar computers connected using a high speed network usually running the same operating system and same software toolset, and more significantly under one administrative domain. It can be loosely defined as a group of computers that work together in a unified manner, such that in many aspects, they appear similar to a single computer. Utilizing such clusters to provide or execute computing solutions is known as cluster computing [1].

Clusters can be classified into three main types: high performance clusters, load balancing clusters and high-availability clusters. High-availability clusters are mainly used for improving the availability of services that the cluster provides. They operate by having additional nodes (computers) to provide service when some of the system components fail. Load balancing clusters operate by accepting all incoming workloads through one or two front-end computers and then distributing the tasks amongst a collection of back end machines. These types of clusters are generally used as server farms. High performance clusters are implemented to provide increased performance by splitting a task into several sub tasks, which are then executed on different nodes in the cluster.

1.2 Grid Computing

Grid computing [2] can be defined as a group of clusters loosely connected together via the internet. The Grid is the next generation in distributed computing. It is a conglomeration of hardware and software resources pooled together to provide huge computing power and storage capabilities.

The key distinction between clusters and grids is mainly in the way resources are managed. In the case of clusters, a centralized resource manager performs the resource allocation and all nodes cooperatively work together as a single unified resource. Whereas, grids are a collection of different clusters hence, resource management is de-centralized. In addition, grids do not aim at providing a single system view.

1.3 Goal of this thesis

A new era has dawned with the advent of HPC. Scientists have exploited HPC to its benefit like no other group. Here, we look at two such implementations of HPC that have broken new ground.

First, a high performance cluster computing implementation for genome biology known as REPCLASS [3]. REPCLASS is an automated software tool that classifies Transposable Elements (TEs) [4] (repetitive DNA) in eukaryotic genomes [5]. There are software tools that help with de novo identification of repeats in the genome but REPCLASS is the first tool that helps to classify these repeats. Classifying TEs in different genomes not only helps biologists study the TE composition of different genomes but also brings them a step closer to a complete understanding of the genome. In this document, we take a detailed look at the improvements that not only make REPCLASS workflow quicker but also help it weed out non-pertinent artifacts and provide better biological information.

The second high performance implementation we consider is a grid computing solution for High Energy Physics (HEP). Panda [6] is a massive grid computing middleware designed to handle enormous amounts of data generated by the particle physics experiment ATLAS [7] located at CERN near Geneva, Switzerland. In particular, we talk about a problem with the centralized approach of Panda to the distribution of pilots to different types of clusters and we present an implementation that alleviates this problem for clusters running Portable Batch System (PBS) [8] by modifying the existing solution to employ a de-centralized approach.

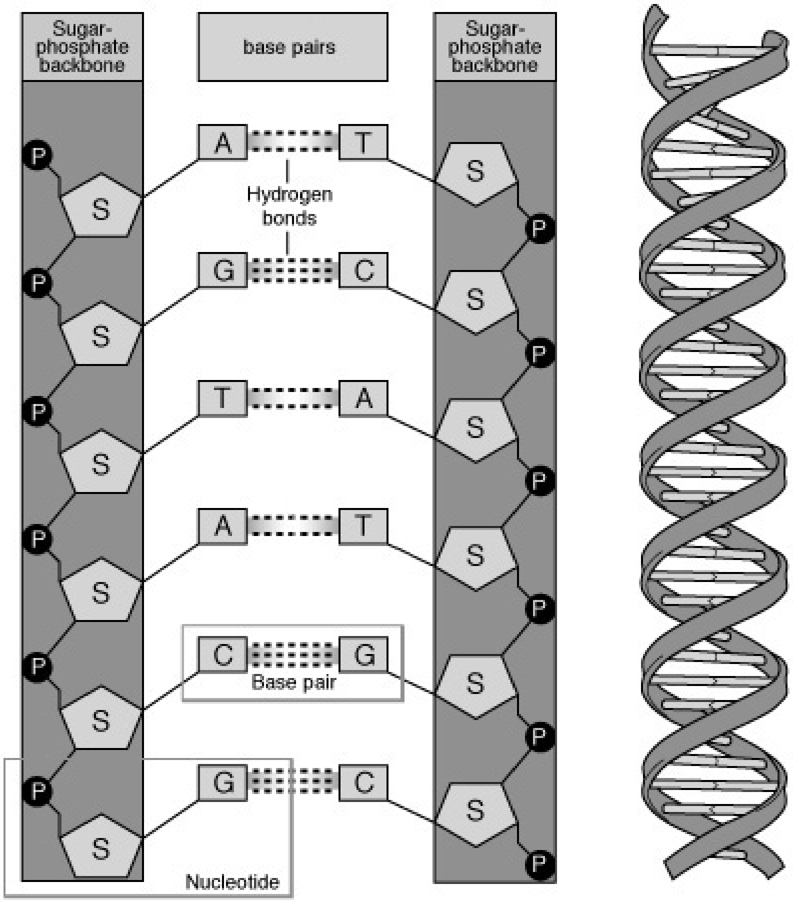
CHAPTER 2

CLUSTER COMPUTING SOLUTION FOR GENOME BIOLOGY

2.1 Background

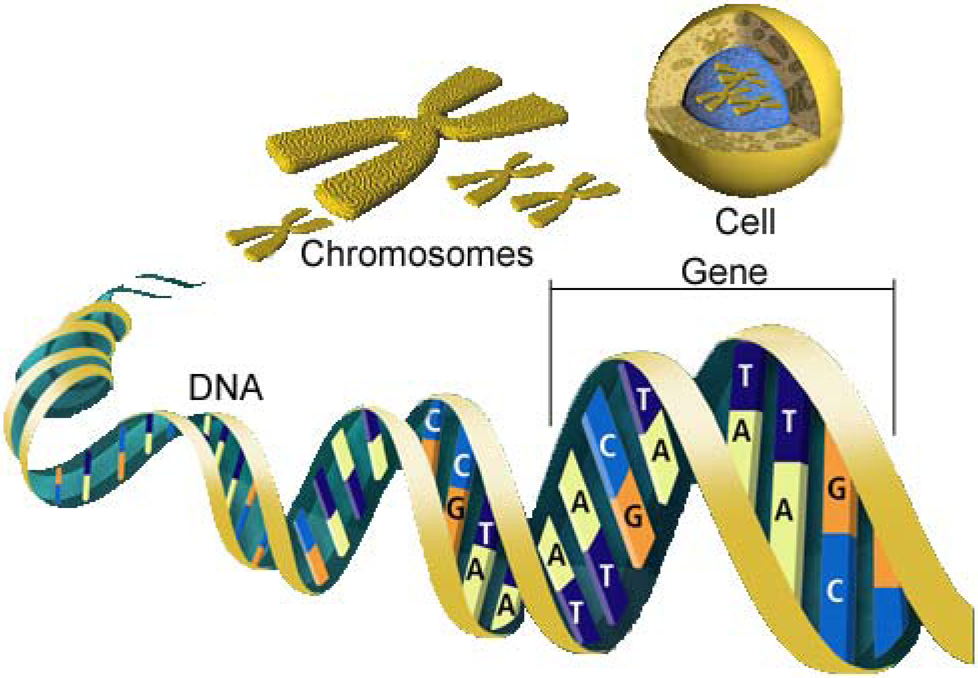
2.1.1 Genome Biology

All living organisms can be classified in two broad categories, prokaryotes and eukaryotes. Prokaryotes (e.g. bacteria) are a group of organisms that lack a membrane-bound structure called nucleus in their cell, the fundamental working unit of every organism. Most prokaryotes are unicellular organisms, but some are multicellular. Eukaryotes (e.g. humans) on the other hand have cells with a well-defined nucleus. The nucleus provides better regulation of bodily functions. It is surrounded by cytoplasm, which mainly consists of water. All the bodily functions are performed by means of proteins that are produced in the cytoplasm. The type of proteins produced in a cell depend on the proteins encoded in the DNA [9] contained within the nucleus of that cell which in turn depend on the genes that are turned on in the DNA which finally depend on the organ that cell belongs e.g. liver cells produce different proteins than that produced by brain cells. DNA, which stands for Deoxyribose Nucleic Acid, is fundamentally made up of a chemical compound known as nucleotide. A nucleotide consists of a molecule of sugar, a molecule of phosphoric acid and a base. This base can be A (Adenine), T (Thymine), C (Cytosine), or G (Guanine). Every nucleotide has a 5' (5 prime) and 3' (3 prime) end.



(Image Credit: U.S. Department of Energy Human Genome Program, http://www.ornl.gov/hgmis)
Figure 2.1: Structure and composition of nucleic acid

A number of nucleotides are linked together to form a nucleotide chain or sequence (e.g. ACCGTCGA) with 5’ end of one nucleotide attached to 3’ end of another. Further, a ladder like structure is formed by combining two nucleotide chains with the hydrogen bonding (pairing) between the bases of respective nucleotides forming the steps (figure 2.1). A binds with T and C binds with G on the opposite strand. A single such pairing of bases is known as a base pair (bp). This ladder like structure is then coiled to form a double helix, which is further coiled to get chromosomes (figure 2.2). Two strands, which form the double helix coil, are referred to as a forward strand and a reverse complement strand going from left to right and right to left in a positive direction. In the positive orientation the start of the strand is called 5’ and the end is 3’.



(Image Credit: U.S. Department of Energy Human Genome Program, http://www.ornl.gov/hgmis)
Figure 2.2: Cell, chromosomes and DNA structure

The complete set of DNA of any organism is collectively known as the genome. Genome of any organism contains both genes and repetitive DNA. Genes are fundamental hereditary subunits. They encode proteins that are synthesized in the cytoplasm to perform bodily functions. Majority of the genome is repetitive DNA with genes forming a very important minority.

The nucleus also contains other molecules of proteins and RNA (Ribonucleic Acid). The function of some RNA is to copy genetic information from DNA and export this information into the cytoplasm where it is translated into proteins. RNA is single stranded whereas DNA double stranded, and RNA contains uracil instead of thymine as one of the possible bases.

There are two general types of genes: non-coding genes and protein coding genes. Non-coding genes encode various functional RNA molecules. Coding genes serve as a template for encoding proteins through a two-step process where the gene is first transcribed into RNA and then translated into amino acid chains. The boundaries of a protein-encoding gene are defined as the points at which transcription begins and ends. The core of a protein coding gene is the coding region, which contains the nucleotide sequence that is eventually translated into an amino acid (and eventually a protein) sequence. The genes are composed of two primary alternating structural components called exons and introns (figure 2.3). The exons carry the information required for protein synthesis and they are translated into the corresponding proteins. The noncoding parts of the gene are called introns.

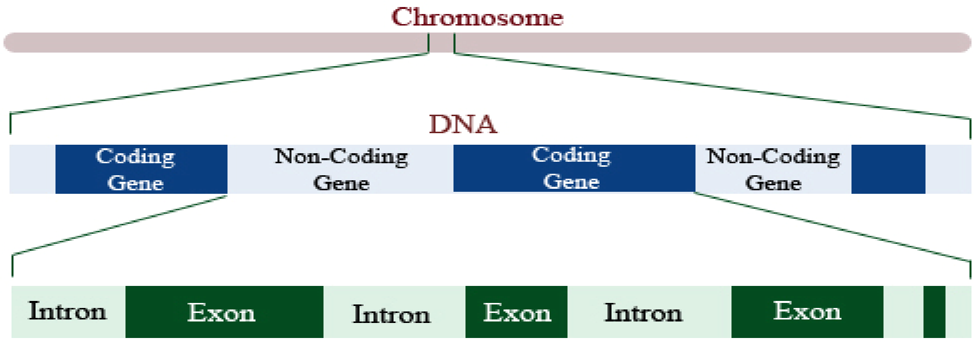


Figure 2.3: Gene, exons and introns representation [3]

The genome size varies for each organism. Table 2.1 shows the genome sizes of some eukaryotic organisms along with the estimated number of genes [10] and percentage content of repetitive DNA. The table illustrates that the variations in genome size cannot be explained by the variation in gene numbers, but that it correlates relatively well to the amount of repetitive DNA. Larger eukaryotic genomes, such as those of some plants or those of mammals contain larger amount of repetitive DNA than the smaller genomes of nematode or yeast, for example.

Table 2.1: Genome size and number of genes [10, 11, 12]

Organism	Genome Size (bases)	Estimated Genes	% repetitive DNA
Human (Homo sapiens)	~3 billion	~30,000	44.4
Lab mouse (M. musculus)	~2.6 billion	~30,000	39
Fruit Fly (D. melanogaster)	~137 million	~13,000	22
Roundworm (C. elegans)	~97 million	~19,000	6

2.1.2 Repetitive DNA

Repetitive DNA was earlier considered junk but, recent studies have revealed that TEs might be involved in various important basic biological processes like gene silencing, gene regulation and the evolution of genes and proteins [13-16]. In most eukaryotic genomes, the largest fraction of repetitive DNA is made of interspersed repeats. The human genome (figure 2.4) contains 44.4% [12] of interspersed repeats, 3% coding regions (excluding those of TEs) and 2% of satellites and micro-satellites (small repeated sequences at least 10 bp long). The remaining are non-coding regions of unknown origins.

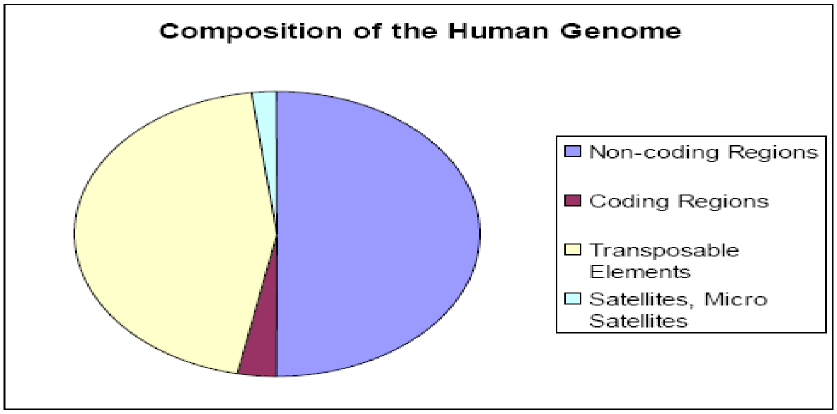


Figure 2.4: Human genome composition [3]

Repetitive DNA in the genome can be classified as either tandem repeats or interspersed repeats. Tandem repeats are sequences that contain multiple continuous repetitions of the same sequence motifs. Interspersed repeats are copies of long nucleotide sequences dispersed at multiple locations throughout the genome. Interspersed repeats are usually much longer than tandem repeats. Almost all the interspersed repeats are generated by a method called transposition and therefore are referred to as Transposable Elements (TEs) or transposons. Transposition is a reaction that facilitates the movement of discrete DNA segments between many chromosomal sites.

Tandem repeats are typically excluded from genome sequence projects because of inherent difficulty in cloning tandem repeats. However, tandem repeats are easily identified and classified based on the pattern or sequence, unit length of the sequence and the number of units. Microsatellites are typically shorter than 1000 bp and composed of tandem repeat units 2bp to 10bp long. Minisatellites are composed of tandem repeat units longer than microsatellites, from a few bp to 100bp. Minisatellites span from 1 kbp to 100 kbp. Satellites are generally made of larger units (>100bp) and occur in arrays of 100-1000 tandem repeat units and are predominantly located in the well-defined chromosomal regions, such as centromeres. In contrast, interspersed repeats are derived from a very diverse range of TEs for which a much more complex classification is required. They have different structural properties and different methods of transposition, by which they replicate. More details on their classification and characteristics are provided in the following sections.

2.1.3 Transposable Elements

Transposable elements are generally 100 to 20,000 bp long. They make up a large part of interspersed repeats in the genome of any eukaryotic organism and have huge diversity. They have different structural properties and different methods of transposition, by which they replicate. TEs transpose themselves to another part of the genome using either the "copy-paste" or the "cut-paste" method. In the cut-paste mechanism, a TE is excised from one site (a position in the DNA sequence) and inserted at another site on either the same or a different chromosome. In the copy-paste mechanism, a TE creates a RNA copy of itself, which is reverse transcribed into a DNA molecule that is subsequently reinserted in the genome (figure 2.5). There are two main classes of TEs based on whether or not they have an intermediate involved during transposition or not, (1) transposition with an RNA intermediate (copy-paste) and (2) direct DNA transposition (cut-paste).

2.1.3.1 Transposition with a RNA intermediate

This class of transposable elements (CLASS I) follow a mechanism of transposition which involves an RNA intermediate. This RNA copy is converted to DNA copy using an enzyme called as reverse transcriptase. Transposition mechanism of these TEs is similar to retrovirus replication hence they are also known as retrotransposons. Some of these TEs have long direct repeats on either ends. These repeats are known as long terminal repeats (LTR). This result in two subclasses: LTR retrotransposons and Non-LTR retrotransposons.

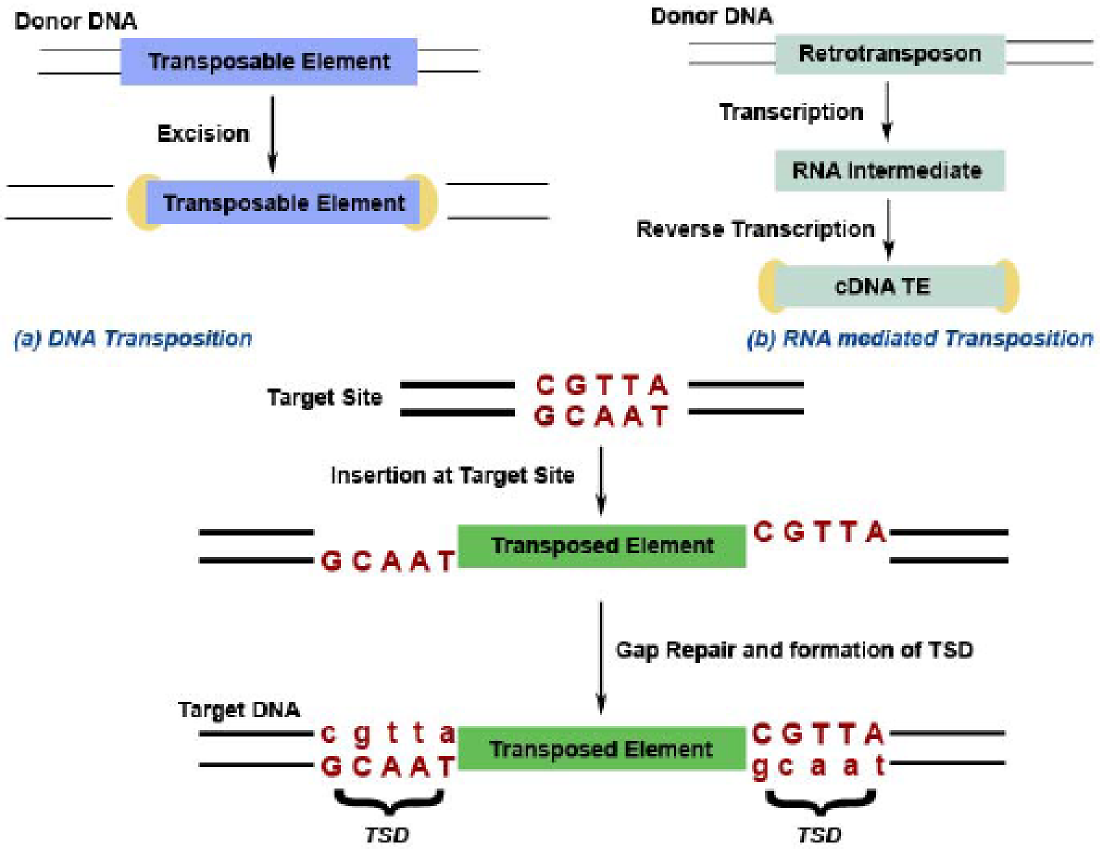


Figure 2.5: Transposition of Transposable Elements (TE) [3]

2.1.3.2 Direct DNA Transposition

Class II elements transpose directly from the DNA and do not form intermediate sequences. This class is subdivided into two major subclasses, DNA transposons and Helitrons. DNA transposons are elements bounded by terminal inverted repeats (TIR) on either end. The DNA transposition [17] creates short gaps on either side of the inserted sequence on the target location. The host site repairs these gaps, creating target site duplications (TSD) (figure 2.5), which is the characteristic of most transposons. The other subclass of elements, Helitrons, transposes in a similar way but they do not form target site duplications. These elements insert between A on the 5' end and T on the 3' end. They also have different structural properties when compared to DNA transposons.

2.1.3.3 TE Classification

The levels of TE classification are based on certain properties of the TEs, which distinguish them. *Class* distinctions are created based on the transposition intermediate and mode of transposition. In the next level, *subclass* is distinguished based on structural properties, integration mechanism and the coding capacity. The structural properties are terminal inverted repeats (TIRs), long terminal repeats (LTRs), and terminal simple sequence repeats (SSRs). The integration mechanism is reflected by the target site duplications (TSD), which the TEs create in the flanking host DNA upon insertion on the target side.

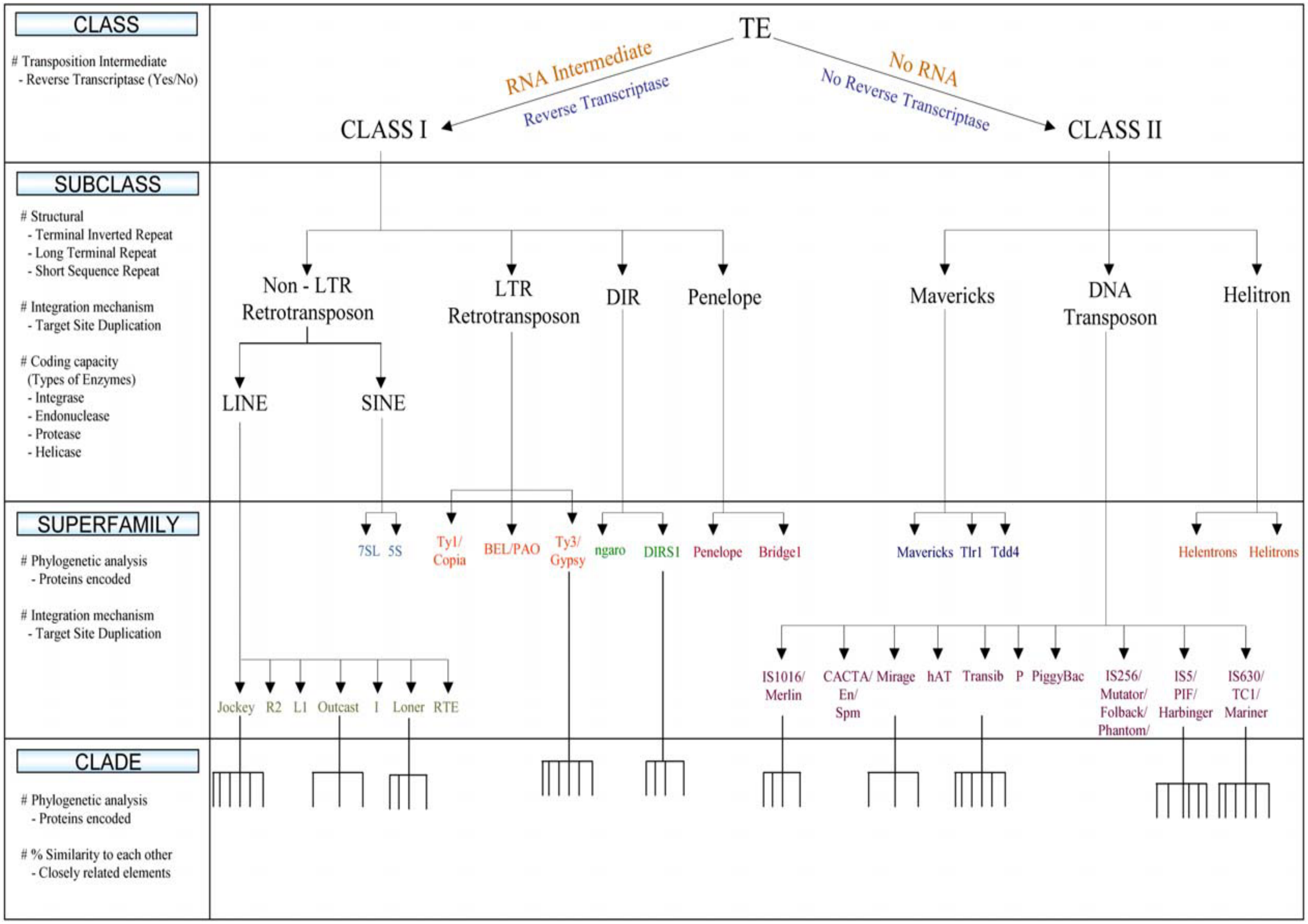


Figure 2.6: TE Classification Chart

The next level of classification is *superfamilies*, which are distinguished by integration mechanism (e.g. size and sequence of the TSD) as well as phylogenetic analysis of the element-encoded proteins (if present). The classification chart [3] (figure 2.6) provides detailed information about the classification of TEs for eukaryotic genomes, along with the classification criteria.

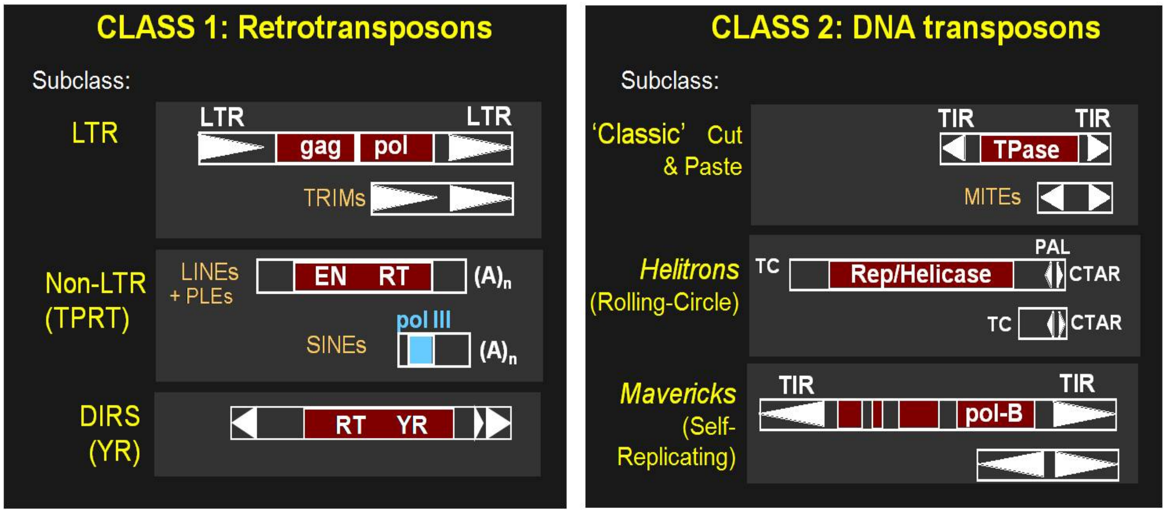
2.1.4 TE Characteristics

Different types of TEs exhibit different peculiar characteristics. Knowledge of presence or absence of these characteristics can be exploited to classify the TEs. We look in detail at some of these characteristics. For a recent review of the classification of TEs, see Wicker et al. (2008) (Nature Reviews Genetics) [43].

2.1.4.1 Homology

Different elements code for different enzymes, which helps us distinguish them. Each group of TEs further consists of autonomous and non-autonomous elements. Autonomous elements encode their own protein-coding domains (transposase, gag, pol, EN, RT, etc), as shown in the figure below while non-autonomous elements do not have protein-coding domains. Non-autonomous elements may nevertheless still propagate by using the enzymes encoded by the autonomous elements. We can classify autonomous elements by identifying the proteins they encode. Sometimes elements can be classified till the depth of superfamilies. Since, non-autonomous elements do not code for proteins, we exploit other structural properties to classify these elements

Most of these families of TEs occur in large copy numbers within a genome. These copy numbers vary depending on the element. The Alu family and all its subfamilies are the most abundant in the human genome, with at least a million copies. Figure 2.7 shows the different classes and some subclasses with their structure. For example, the cut-and-paste DNA transposons are flanked by terminal inverted repeats, while the LTR retrotransposons by direct long terminal repeats. The non-LTR retrotransposons do not have structural motifs, but instead feature simple sequence repeats at their 3' end (see definition below)



(Image Credit: Dr. Cedric Feschotte)
Figure 2.7: Structure of some TE subclasses

2.1.4.2 Target Site Duplications (TSD)

TSDs are formed during the insertion of a TE into a target site, which is typically associated with a staggered DNA double strand breaks. This double stranded gap is then repaired by the host genome to form target site duplications. The formation of a TSD is shown in figure 2.5. It has been found that these TSDs are sometimes conserved in length and in certain cases, a clear sequence preference for the TSD may be observed.

However, not all TEs create TSDs upon insertion. Using TSD information, we can classify most TEs to the depth of superfamilies. Following are some of the examples:

- 4-6bp TSDs for most LTR retrotransposons
- ‘TA’ TSD sequence for the Tc1/mariner Superfamily (DNA Transposons)
- TSDs 8bp long for the hAT Superfamily (DNA Transposons)

2.1.4.3 Terminal Inverted Repeats (TIR)

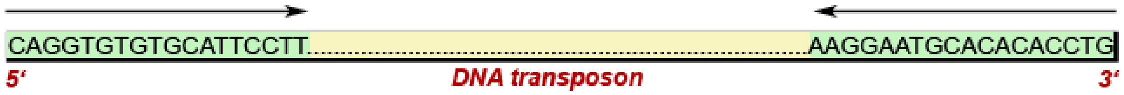


Figure 2.8: Terminal Inverted Repeats (TIR)

An inverted repeat is one where two different segments of the double helix read the same but in the opposite directions. Terminal inverted repeats are the inverted repeats that occur at the ends of a transposable element. This structure is common in all DNA transposons. The length of the TIRs varies from 10bp-500bp.

2.1.4.4 Long Terminal Repeats (LTR)

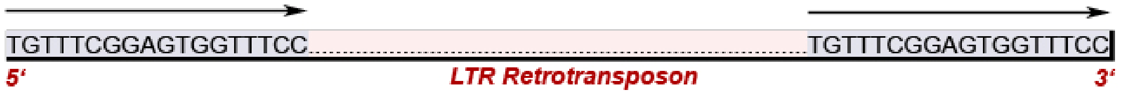


Figure 2.9: Long Terminal Repeats (LTR)

Long terminal repeats are long direct repeating sequences of DNA that occur on either end of the TE. LTRs are a structural feature of LTR retrotransposons. LTRs vary in length from 100bp to several kbp. For illustration, only a short stretch of an LTR is shown, in direct orientation, figure 2.9.

2.1.4.5 Simple Sequence Repeats (SSR)

SSRs are small sequences of 2bp-10bp length, which repeat in constant intervals. They are mainly characteristic of satellites, SINEs (Short Interspersed Nucleotide Elements) and LINEs (Long Interspersed Nucleotide Elements). They occur in the flanking regions or within the element, but exist either on the 3’ or 5’ end, but not on both ends.

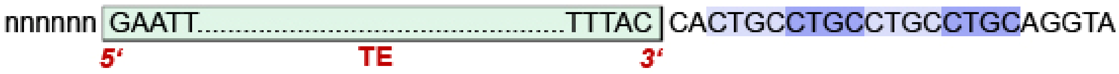


Figure 2.10: Simple Sequence Repeats (SSR)

2.1.5 Genome Sequence Analysis

Whole genomes for new species are being sequenced at an ever-increasing pace. There are currently around 4000 active genome project, out of which around 1000 are eukaryotic genomes [44]. These generate enormous amounts of raw sequence. Thus, there is an urgent need to annotate and analyze the content of these sequenced genomes in order to get useful biological information. Scientists hope that comparative analysis of these genomes will help understand how the genome works and evolves as a whole, and how the genes work to regulate the growth, development and maintenance of an organism. Genes, which form only a minority of most eukaryotic genomes, have been studied and continue to be studied extensively. However, repetitive DNA, which is mainly comprised of diverse set of TEs, forms a major part of the eukaryotic genome, has not been explored as thoroughly. TEs are the fastest evolving fraction of the genome and one that closely co-relates to the genome size. Once considered as ‘junk DNA’, it has now been established that TEs play a significant role in gene regulation and genome evolution. Hence, TEs need to be studied much more closely to clearly understand their role in genome structure and function. There are various tools available for analyzing genes but comparatively, there are very few tools that can help in analyzing repetitive DNA. One tool that goes a long way in helping with the analysis of TEs is REPCLASS.

2.2 REPCLASS

REPCLASS [3] attempts to classify newly identified TEs taking into account various characteristics, which characterize a repetitive element. REPCLASS has been predominantly developed for the classification of repeats identified in complete genome sequences and grouped into families by de novo identification programs like RECON [18] or RepeatScout [19]. Even though REPCLASS will accept any sequence as input, it is essential that the input is a consensus of the intact and ancestral (pre-mutated) sequence of an active repeat element, containing structural and sequence information necessary for the classification. In an ideal case de novo identification is performed on a new genome and the complete repeat library consisting of the consensus sequences of the repeat elements is provided as input to REPCLASS.

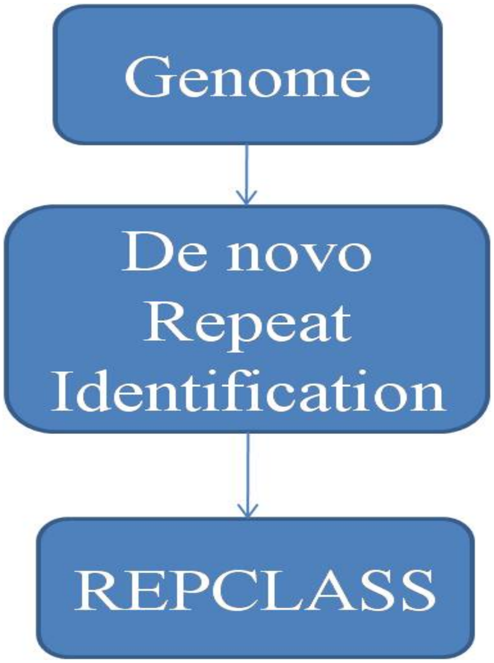


Figure 2.11: REPCLASS overview

2.2.1 De novo Identification of Repeat Families

Identifying families of repetitive elements is a vast and complex algorithmic problem. There are only a few algorithms and programs performing de novo identification of repeat families in whole genome sequences, and packaged as software tools such as RECON, RepeatScout, ReAS [20] and Piler [21]. In order to generate the input the consensus library from complete genomes REPCLASS uses RepeatScout, this is one of the methods to identify repeat families de novo. In this study, we used RepeatScout (RS) to produce de novo repeat library. RS has been shown to provide the best compromise between quantity and quality among de novo repeat finder programs [45]. However, any other program can be used to generate an initial list of TE sequence consensuses for a given genome to be analyzed by REPCLASS.

The complete genome sequence is the input to RepeatScout, which outputs a fasta [22] format file of the consensuses of the identified repetitive elements. RepeatScout reports all kinds of repetitive elements, including tandem repeats, satellites, micro-satellites and TEs. Since, they clash with the use of SSRs to classify non-LTR elements, all repeats predominantly or entirely composed of tandem repeats, SSR and other low-complexity repeats need to be filtered out. REPCLASS use Tandem Repeat Finder (TRF) [23] and nseg [24] programs to filter out these elements

2.2.2 Automated REPCLASS workflow

There is no one method that can be used to classify all the TEs with all its diversity. Hence, REPCLASS uses three different methods to classify TEs. REPCLASS workflow consists of four main parts

- Homology based classification
- TSD based classification
- Structural based classification
- Validation and grouping of results

A fasta format file is an input to the workflow and is passed on to each of the three methods. These methods are executed in parallel to exploit the inherent parallelism of the solution and helps achieve better computing performance. Finally, results from all these methods are grouped together and outputs a tentative classification for the TEs.

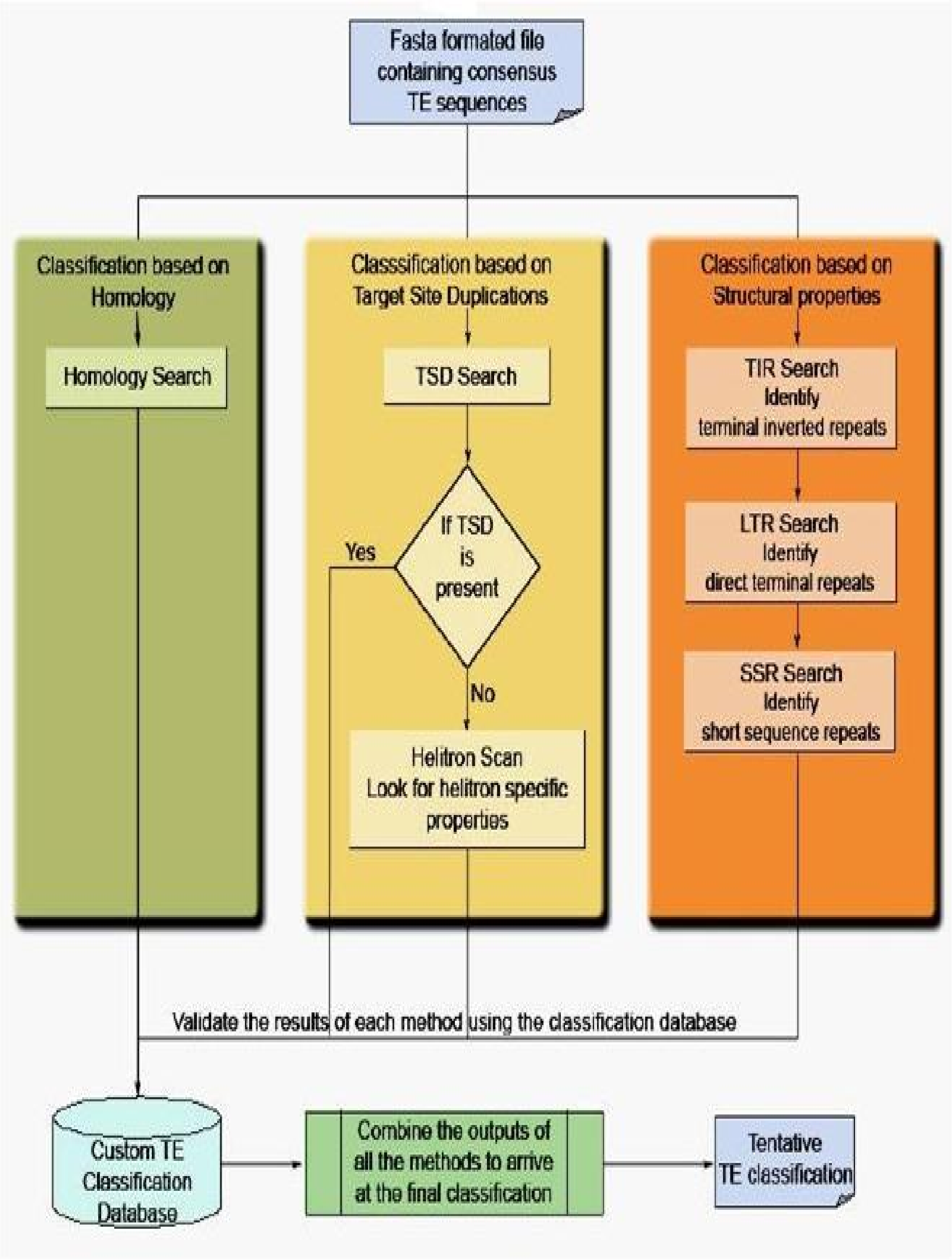

Fasta formated file containing consensus TE sequences
Classification based on Homology
Homology Search
Classsification based on Target Site Duplications
TSD Search
If TSD is present
Yes
No
Helitron Scan Look for helitron specific properties
Classification based on Structural properties
TIR Search Identify terminal inverted repeats
LTR Search Identify direct terminal repeats
SSR Search Identify short sequence repeats
Validate the results of each method using the classification database
Custom TE Classification Database
Combine the outputs of all the methods to arrive at the final classification
Tentative TE classification

Figure 2.12: REPCLASS classification workflow [3]

2.2.3 Homology Search

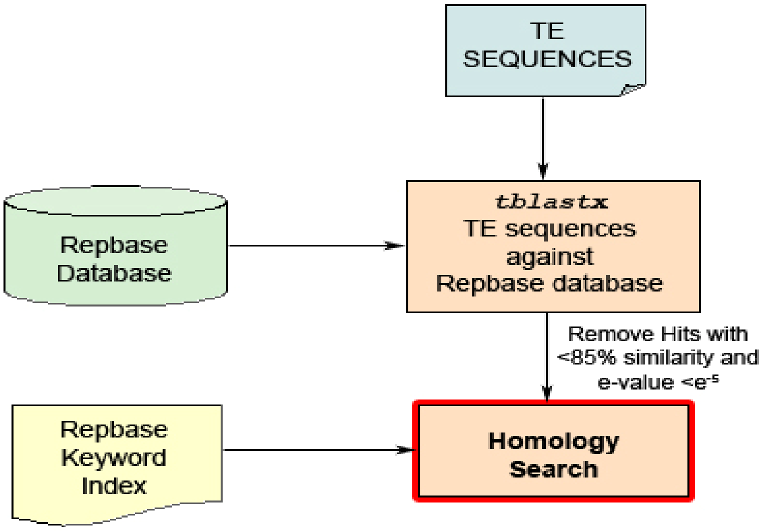

TE SEQUENCES
Repbase Database
tblastx
TE sequences against Repbase database
Remove Hits with <85% similarity and e-value <e-5
Repbase Keyword Index
Homology Search

Figure 2.13: Homology Based Classification

The homology search uses tblastx to compare proteins encoded by the input TE families (query) to proteins encoded by known TE families (database). The tblastx script, which is part of the WU-BLAST [46] package, performs a heuristic search for local alignments of the protein sequences and detects regions of similarity between the query and the sequences in its database. We get information about known TE families from a database known as Repbase Update [47]. Repbase is a manually curated database of repeats (interspersed and tandem) found in eukaryotic genomes. The Repbase database is the most authoritative repository of known TE classifications and sequences.

2.2.4 TSD Search

Target site duplication is the process of creating duplicate sequences of a portion of the host genome at the ends of the transposable element when they are inserted into new locations within the genome. This process creates the same sequence on both ends of the transposable element (for example, CGTTA<TE sequence>CGTTA, also see figure 2.5). Different families of TEs form different TSDs during insertion and this helps distinguish the TE family. This step is a computationally and memory intensive task as the entire genome needs to be searched for the various copies of the input TE sequence.

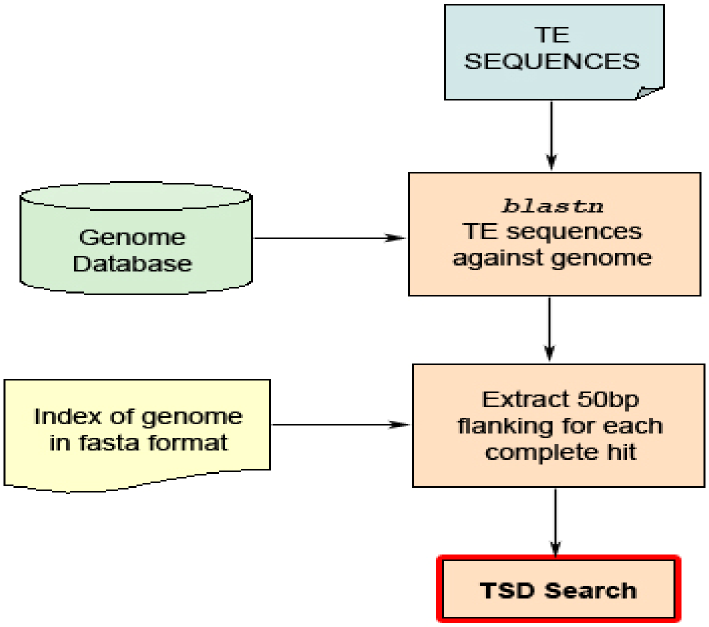

TE SEQUENCES
Genome Database
blastn TE sequences against genome
Index of genome in fasta format
Extract 50bp flanking for each complete hit
TSD Search

Figure2.14: TSD Search

2.2.5 Structural Classification

The final method is the structural search, which is designed to identify the structural characteristics of TE subclasses and superfamilies. This is also a computationally intensive search since all the possibilities for such structures need to be searched.

2.2.5.1 Helitron Search

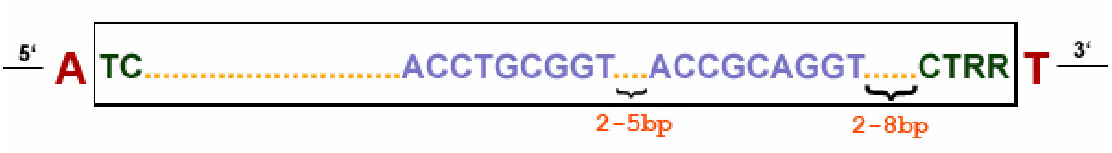


Figure 2.15: Helitron Structure

The Helitron [25, 26] subclass exhibits some unique structural properties that distinguish it from other repeated elements. All elements belonging to this class have conserved 5'-TC and CTRR-3' (R = A or G) ends which do not have terminal inverted repeats. They contain 16-20bp long palindromes separated by 10-12bp from the 3' end and transpose precisely between the 5'-A and T-3', with no modifications at the AT target sites and the palindrome is rich in GC content.

2.2.5.2 LTR Search

Long Terminal Repeats (LTR) or direct repeats are a set of sequences that repeat in the same direction at both the 5' and 3' ends. LTRs vary in size from 100bp to several thousand bp. REPCLASS uses the sliding window algorithm to look for these repeats. It starts with 50 bp long windows on either sides of the sequence and allows a mismatch of 1bp every 10 bps.

2.2.5.3 TIR Search

Terminal inverted repeats (TIRs) are characteristic of DNA transposons. They are inverted repeats ranging for 10bp to 500bp, on either ends of the transposable element sequence. The presence of a terminal inverted repeat confirms that the corresponding TE belongs to the DNA transposon subclass of class II. REPCLASS TIR search uses a third party einverted script, which is a part of the Emboss [27] suite to find all the inverted repeats. These inverted repeats are not necessarily terminal inverted repeats. Hence, TIR search parses through the einverted output to find inverted repeats that appear within 30bp on either end of the sequence.

2.2.5.4 SSR Search

Simple Sequence Repeats (SSRs) are formed by multiple repetitions of the same basic sequence motif. SSRs are characterized by their base sequence, the length of the sequence and the number of units. These parameters vary significantly, with SSRs from 1bp to 10 bp long. Repeats with more than 10bp sequence are considered as tandem repeats, satellites or micro-satellites. REPCLASS is only interested in SSRs that occur only at one end of the input sequence. REPCLASS SSR search is a two-step process. It looks for poly A tail and simple sequences repeats.

2.2.5.4.1 Poly A Tail Search

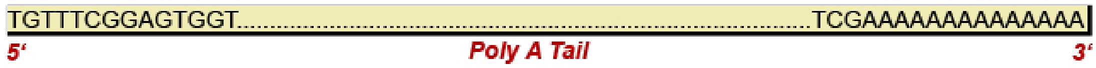


Figure 2.16: Poly A Tail

The SSR search scans for Poly A tails at the 3' end of the sequence allowing for a lapse of 10bp from the end and searching the 3' end flanking. This is done considering that the RepeatScout might not have defined the ends properly.

2.2.5.4.2 Simple Sequence Search

This search is based on a sliding window algorithm, where the window size is set from 1 to 5. This search is performed on only hundred base pairs towards the end along with 50bp of flanking.

2.2.6 Validation and Grouping of Results

The final process in the workflow is the grouping of the results obtained by the three methods of classification. The classification results from all the methods (Homology, TSD, and Structural) are combined together to provide a tentative classification. During the combination, some elements may be classified by more than one method. These elements are further checked to verify that they provide the same classification in all the methods. Some of the ambiguities are cleared based on the accuracy of the information gathered by the respective method. In cases where this cannot be resolved, the result of the individual method is shown to the user who can decide on the classification. The final classification for each element contains complete details about the element, such as the target site duplication length and consensus, TIRs, LTRs and SSRs.

2.3 Results Obtained using REPCLASS

2.3.1 Classification of TEs Annotated in Repbase

The classification of previously annotated TEs in Repbase acts as a control for the efficiency and accuracy of REPCLASS. We ran these control experiments for two genomes, *Caenorhabditis elegans* and *Drosophila melanogaster* for the following reasons,

- *C. elegans* and *D. melanogaster* are the most extensively studied species for repeats. The staff at Repbase, a manually curated database for repeats, has over the last decade compiled a comprehensive list of repeats and carefully annotated them. .
- The genome of these two species consists of an assortment of TEs. This provides an understanding of the performance of REPCLASS on a wide variety of TEs.

Before running REPCLASS, we performed the following steps to generate a dataset of consensus sequences from the manually classified TEs of these genomes.

- Since, Repbase has a list of all kinds of repeats even those that are not TEs, all the simple repeats, tandem repeats, satellites and micro-satellites were removed from the Repbase database for both the genomes.
- Since, both these genomes have already been manually annotated in Repbase, all the information about these genomes was removed from Repbase. This was done to prevent all the TEs being classified by REPLASS based on homology.

2.3.1.1 Caenorhabditis elegans

- Number of repeats in Repbase: 174
- Number of repeats after removing unclassified repeats: 116
- Number of repeats classified by REPCLASS: 107
- Percentage classified: 92%

- Accuracy of classification: 96%

The accuracy of classification is measured by matching the superfamily of classification provided by REPCLASS to the superfamily of classification detail in Repbase. From the above numbers we can conclude that REPCLASS could not classify nine repeats that are already annotated in Repbase. REPCLASS failed to classify one LTR retrotransposon, two Helitrons, five DNA transposons and a Non-LTR retrotransposon. The table below shows the distribution of the number and percentage of repeats classified by each method.

Table 2.2: Split of classification by different methods for *C. elegans*

Classified by	No. of repeats classified	% of total classified
TSD + Structural	28	26.17%
Structural	25	23.36%
Homology	21	19.63%
TSD	12	11.21%
TSD + Structural + Homology	10	9.35%
Homology + TSD	6	5.61%
Homology + Structural	5	4.67%

In table 2.2, we can observe that, Structural classified most repeat families at 63.55%, followed by TSD (52.34%) and least by Homology (39.26%). This validates the fact that, the *C. elegans* genome contains significantly more non-autonomous TEs as compared to the autonomous TEs (section 2.1.4.1).

2.3.1.2 Drosophila melanogaster

- Number of repeats in Repbase: 225
- Number of repeats after removing unclassified repeats: 144

- Number of repeats classified by REPCLASS: 140
- Percentage Classified: 97%
- Accuracy of classification: 96%

In the *D. melanogaster genome*, REPCLASS was not able to classify a couple of LTR retrotransposons, a DNA transposons and a Non-LTR retrotransposon. The table below shows the distribution of the number and percentage of repeats classified by each method.

Above results, clearly show that given well-defined repeat consensus sequences REPCLASS was accurately able to classify more than 90% of the repeat sequences. Since, REPCLASS relies heavily on structural properties to classify the TEs, it is very important to have reasonably well defined ends for the input consensus sequences.

Table 2.3: Split of classification by different methods for *D. melanogaster*

Classified by	No. of repeats classified	% of total classified
Homology	79	56.43%
Homology + Structural	45	32.14%
Homology + Structural + TSD	9	6.43%
Homology + TSD	3	2.14%
Structural + TSD	3	2.14%
TSD	1	0.72%
Structural	0	0.00%

In table 2.3, we can observe that, most repeat families (97.14%) were classified using Homology, followed by Structural (38.57%) and then TSD (11.43%). This validates the fact that, the *D. melanogaster* genome contains significantly more autonomous TEs as compared to the non-autonomous TEs (section 2.1.4.1).

2.3.2 *De novo Classification of TE Repeats*

In this step we compare the repeats identified de novo by RepeatScout in the *C. elegans* and *D. melanogaster* genomes and classified by REPCLASS with the repeats compiled and classified by Repbase. This provides an estimation of the performance of RepeatScout and REPCLASS combined and might lead to the discovery and classification of new TE families.

2.3.2.1 C. elegans:

- Number of repeats identified by RepeatScout: 1721 (after TR and nseg filtering)
- Number of repeats classified by REPCLASS: 301
- Percentage Classified: 17%

2.3.2.2 D. melanogaster:

- Number of repeats identified by RepeatScout: 1812 (after TR and nseg filtering)
- Number of repeats classified by REPCLASS: 823
- Percentage Classified: 45%

We can notice that in the aforementioned cases only 116 and 148 TE families have been annotated for *C. elegans* and *D. melanogaster* in Repbase whereas RepeatScout identified 1721 and 1812 repeat families respectively. Thus, there is a big difference in the input libraries in the earlier control experiments and those generated de novo using RepeatScout. This is the case, because the input library generated contains non-TE artifacts such as segmental duplications, gene families. In addition, single TE consensus sequences might have been fragmented into multiple sequences. More interestingly, there might be some TE families that have not annotated by Repbase. This excessive number of repeat families in the de novo input library leads to two problems. First, it leads to some false classifications. Second, it takes much more computation time to process and classify these repeat families.

2.4 REPCLASS Revised

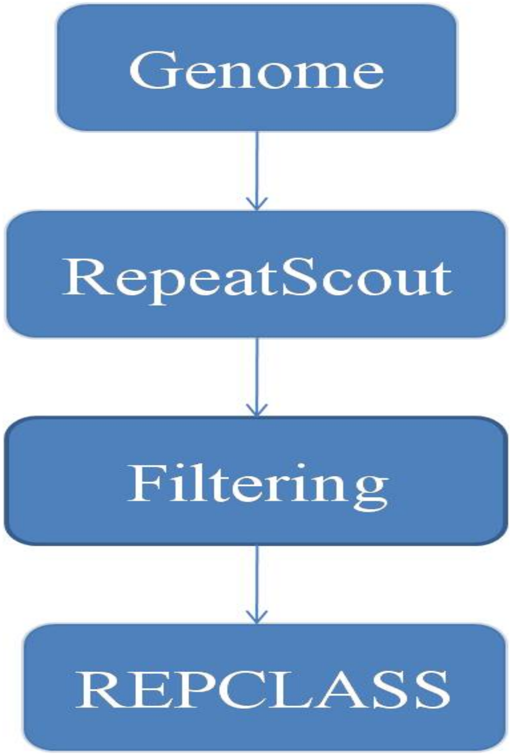


Figure 2.17: REPCLASS 2.0 Overview

To solve the problems mentioned above we decided to filter the RepeatScout output before feeding it into the REPCLASS workflow. We filter the RepeatScout output based on the information about TEs from the literature and empirical observations. The filtering criteria we use are the repeat length of the consensus repeat families and the number of copies per repeat family in the genome. Figure 2.17 shows the revised overview of this new REPCLASS version, REPCLASS 2.0.

2.4.1 Filtering Based on Repeatlength

We calculate the repeatlength by counting the number of nucleotides that form the consensus sequence. Figures show the repeatlength distribution for *C. elegans*.

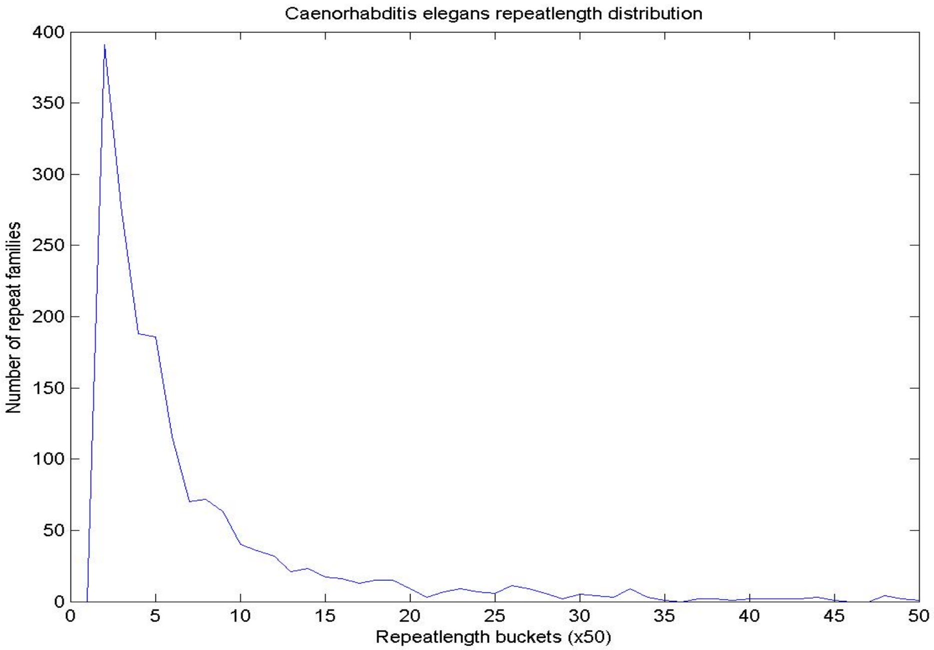


Figure 2.18: *C. elegans* repeatlength distribution

From the literature, we know that most TEs are more than 100 bp long. Also, we observed that all of the TEs annotated in Repbase for *C. elegans* (10 TEs less than 200 bp) and *D. melanogaster* (2 TEs less than 200 bp) are also more than 100 bp long. Hence, we decided to remove all those repeat families identified de novo by RepeatScout that have repeatlength less than or equal to 100 bp. We consider this 100 bp threshold to be the minimum threshold that should be applied to all genome irrespective of the genome size and number of consensus repeats. However, it is possible that a higher threshold might perform better for some other genomes. Hence, we decided to provide a repeatlength distribution as part of the output, for every genome analyzed. This would help the user to adjust the minimum length threshold to better suit the genomic landscape being analyzed.

2.4.2 Filtering Based on Copynumber

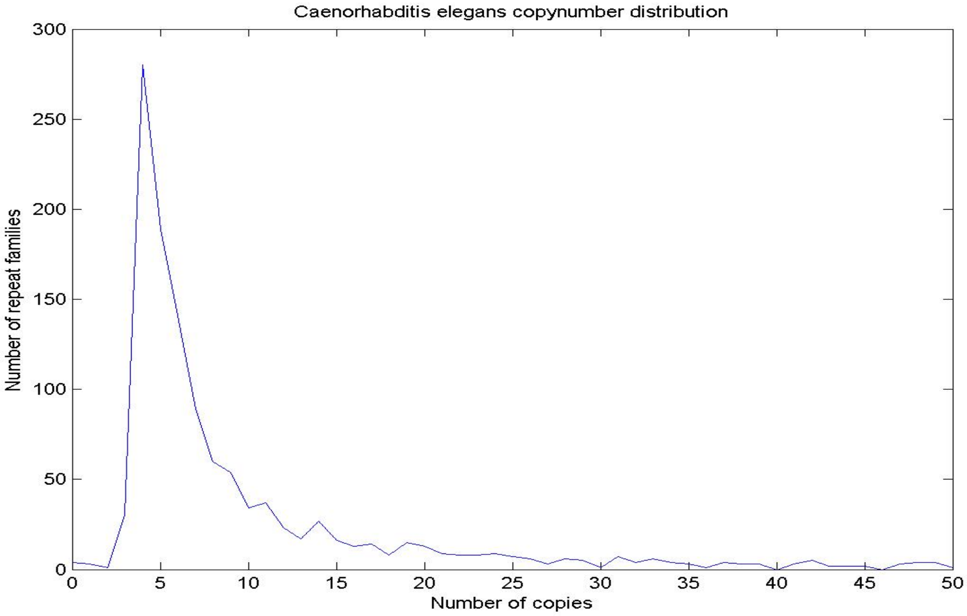


Figure 2.19: Caenorhabditis elegans copynumber distribution

Calculating the copynumber of repeat families is not an easy task. We use the blastn script (part of the WU-BLAST package) to BLAST the query consensus library against the targeted genome. We count all those hits has valid copies that are 85% similar to at the least half of the query sequence. We then draw the distribution graph of both repeatlength and copynumber for the consensus library. Figures show copynumber distribution for *C. elegans*.

2.5 Results Obtained using REPCLASS 2.0

2.5.1 De novo Classification of TE Repeats

2.5.1.1 C. elegans

RepeatScout identified a total of 1851 consensus sequences. Out of these, 130 were filtered out using tandem repeat filter and removing satellites and micro-satellites using nseg. We then plotted the repeat length distribution for the remaining 1721 repeats. After discarding consensus sequences that were less than or equal to 100 bp long and after removing all the repeat families that had less than or equal to 10 copies, we were left with a library of 428 repeat families.

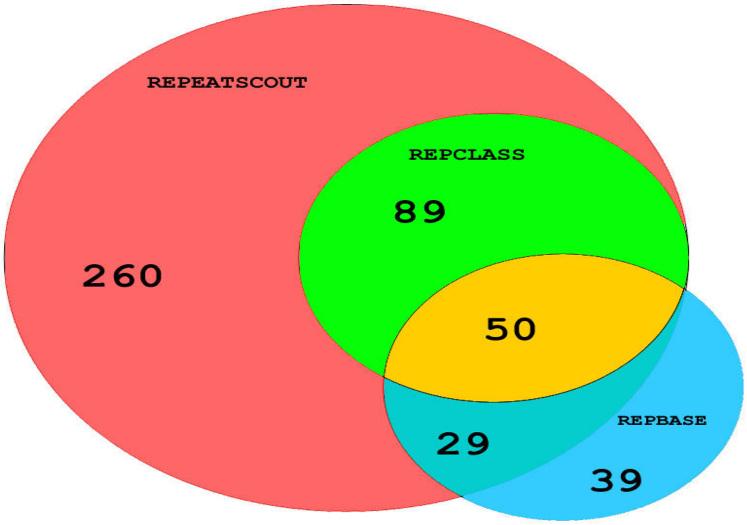


Figure 2.20: Classification distribution for *C. elegans* repeats identified de novo using RepeatScout

139 repeats (32.5%) of the filtered RepeatScout output was classified by REPCLASS. Out of the 428 repeat consensuses, 79 repeats matched Repbase and 50 of

these were classified by REPCLASS. Given the high efficiency of REPCLASS on the *C. elegans* Repbase library (see section 2.3.2.1), we attribute the failure of REPCLASS to classify 29 repeats to incomplete or inaccurate definition of the corresponding consensus sequences by RepeatScout. In addition, 39 repeats are catalogued in Repbase but were not identified by RepeatScout. Finally and most interestingly, 89 of the identified repeats identified by RepeatScout and classified by REPCLASS do not show any significant similarity with repeats in Repbase. Thus, these 89 repeats potentially represent new families or subfamilies. Out of these 89 repeats, 50 were chosen randomly for closer, semi-manual inspection and 43 could be validated as new families and subfamilies correctly classified by REPCLASS [50]. The remaining 289 repeats identified by RepeatScout do not match any sequences deposited in Repbase could not be classified by REPCLASS (figure 2.20). These families might not represent segmental duplications, gene families or other types of interspersed repeats and TEs currently not classifiable by the modules implemented in REPCLASS.

2.5.1.2 D. melanogaster

After running the *D. melanogaster* genome through RepeatScout, we obtained 1844 repeat consensus sequences. After removing tandem repeats, satellites, consensuses with less than 100 nucleotides in length and less than 10 copies in the genome, we were left with 538 repeat families. Figures 2.22 and 2.23 show the repeatlength and copynumber distribution respectively for the *D. melanogaster* genome.

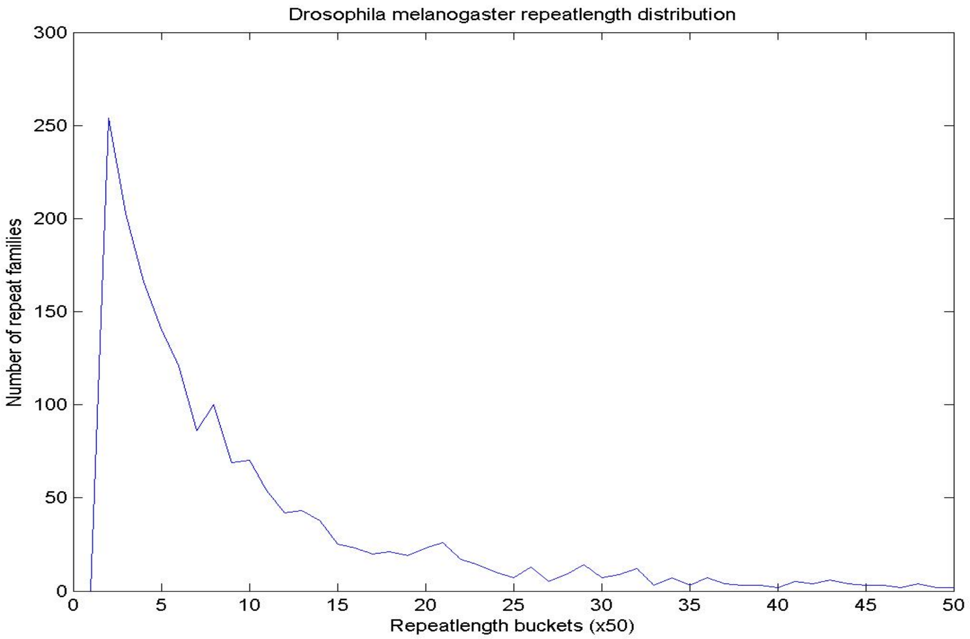


Figure 2.21: *D. melanogaster* repeatlength distribution

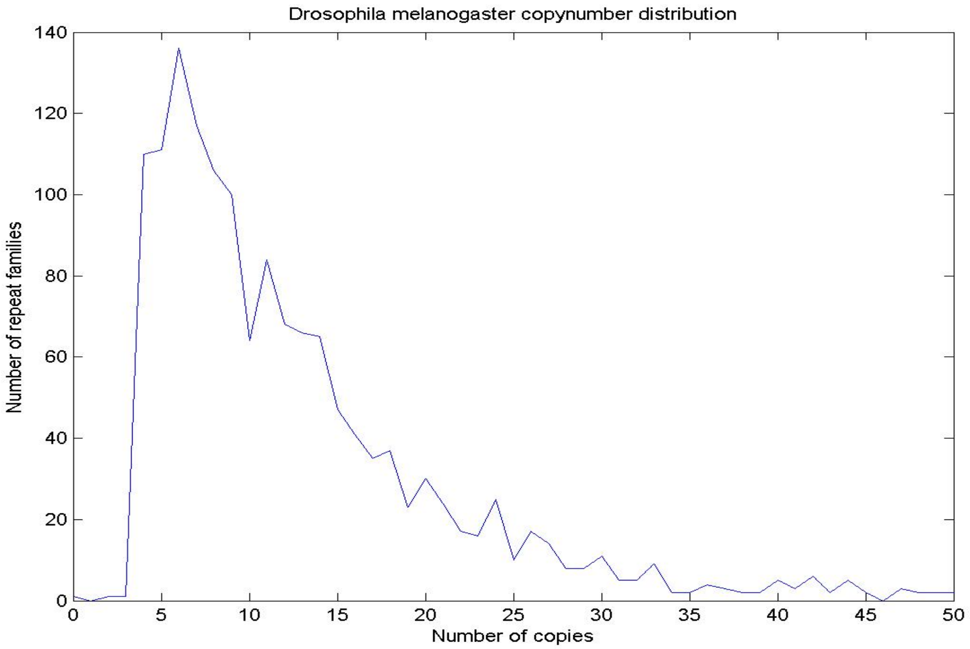


Figure 2.22: *D. melanogaster* copynumber distribution

191 repeats (35.50%) were classified by REPCLASS. Out of the 538 repeats consensuses identified, 92 matched those annotated in Repbase. Out of these 92 repeats, 86 were classified by REPCLASS. 56 repeats that are part of Repbase were not identified by the de novo method. 105 of the identified repeats were classified by REPCLASS but do not show any significant similarity with repeats in Repbase. These 105 repeats potentially represent new families or subfamilies. 341 repeats remain unclassified (figure 2.23).

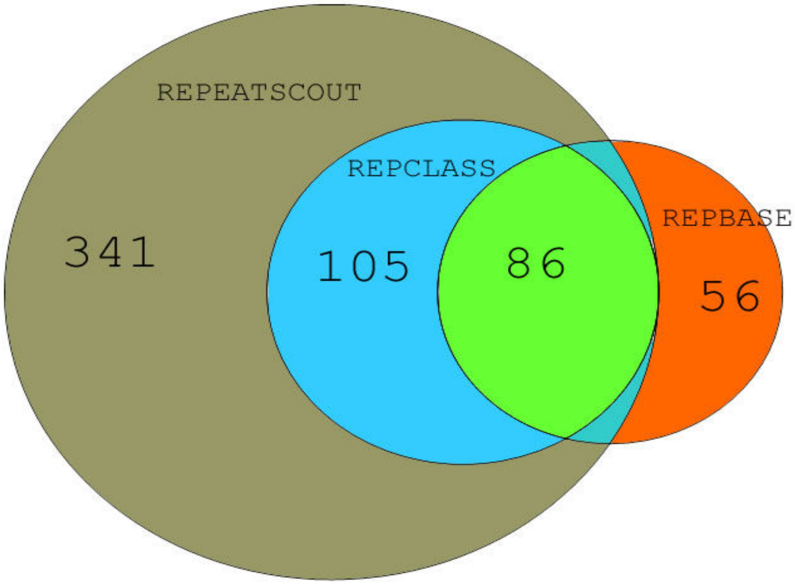


Figure 2.23: Classification distribution for *D. melanogaster* repeats identified de novo using RepeatScout

Above results shows that REPCLASS combined with RepeatScout can be used as a powerful tool for the discovery of new repeat families. De novo classification for C.elegans lasted for about 63 minutes. 28 minutes for filtering and just more than 35 minutes for TE classification. TE classification for D.melanogaster required 2 hours and 10 minutes with additional 38 minutes for filtering before the classification.

2.5.2 Genome exploration with REPCLASS 2.0

In this section, we apply the RepeatScout/REPCLASS suite for the analysis of *Caenorhabditis brenneri* and *Drosophila pseudoobscura* genomes for the following reasons:

- These genomes have never been mined for TEs and annotated thoroughly before. Hence, it provides us with an opportunity to assess how REPCLASS performs on a newly explored genome with lower quality sequence and assembly.
- These species are genetically related to *C. elegans* and *D. melanogaster*, but sufficiently divergent to expect virtually no overlap in sequence between their respective TE populations Also we wondered to what extent the overall TE content and composition would be evolutionarily conserved within the *Caenorhabditis* and *Drosophila* lineages.

2.5.2.1 Caenorhabditis brenneri

RepeatScout identified 8980 repeat families for *C. brenneri*. After removing tandem repeats and satellites using TRF and nseg we were left with 8802 repeat families. Filtering on repeatlength we removed 2506 consensus sequences form the library that were less than or equal to 100 bp long. An additional 2904 repeat families were discarded when we filtered out sequences with less than or equal to five copies in the genome. Figure 2.24 and 2.25 show the repeatlength and copynumber distribution for *C. brenneri* respectively.

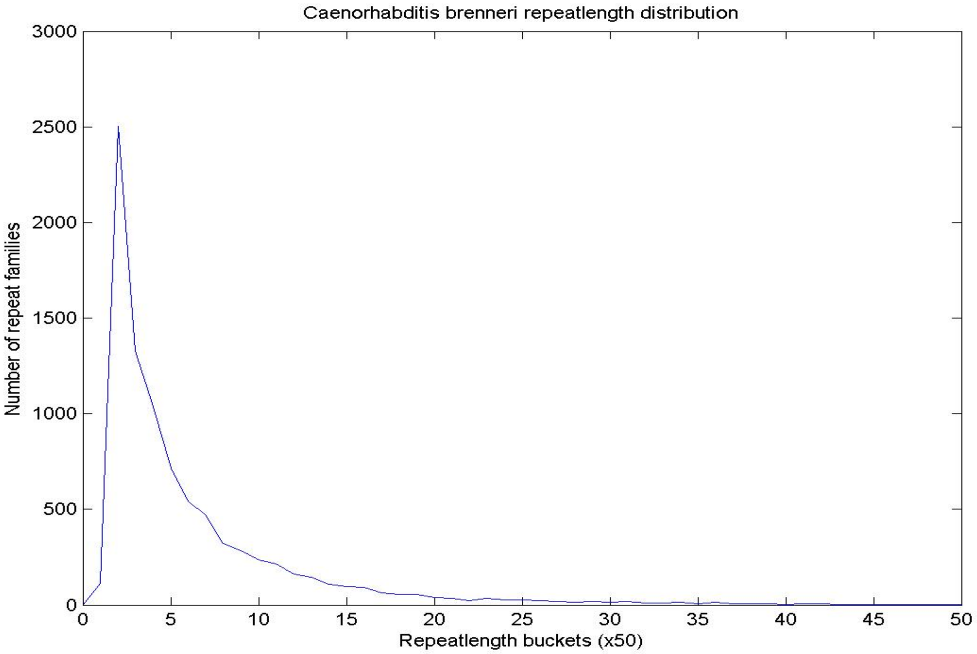


Figure 2.24: *C. brenneri* repeatlength distribution

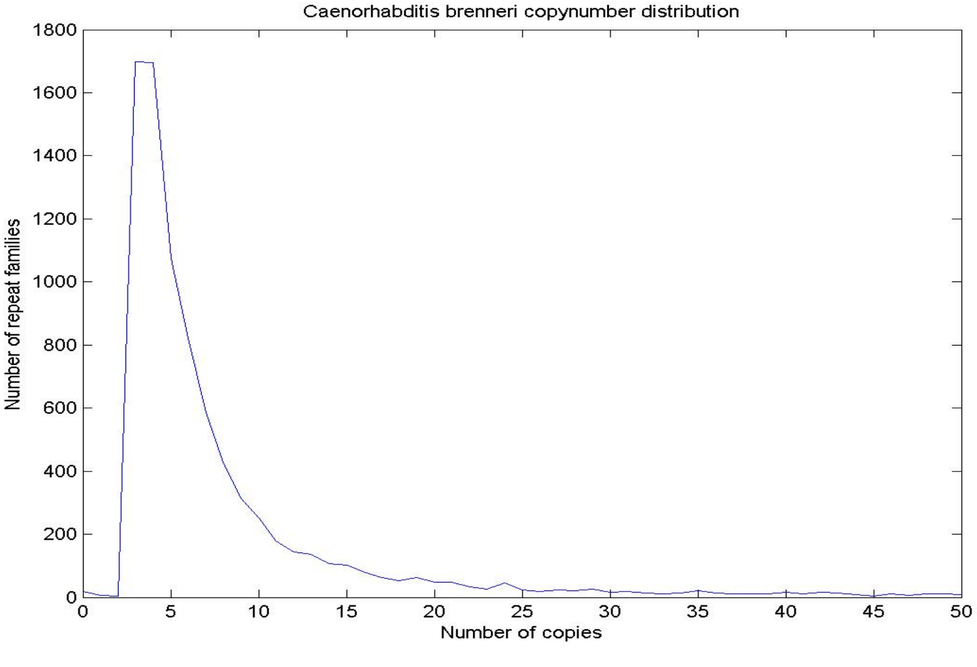


Figure 2.25: *C. brenneri* copynumber distribution

Out of the 3392 repeat families identified de novo by RepeatScout left in the consensus library after various filtering steps, 515 (15.2%) repeat families were classified by REPCLASS. Therefore, we were able to annotate more than 500 different TEs in a genome that had never been studied before. Moreover, we were able to do this extremely quickly, within a couple of hours.

2.5.2.2 *C. elegans* and *C. brenneri* TE Profile Comparison

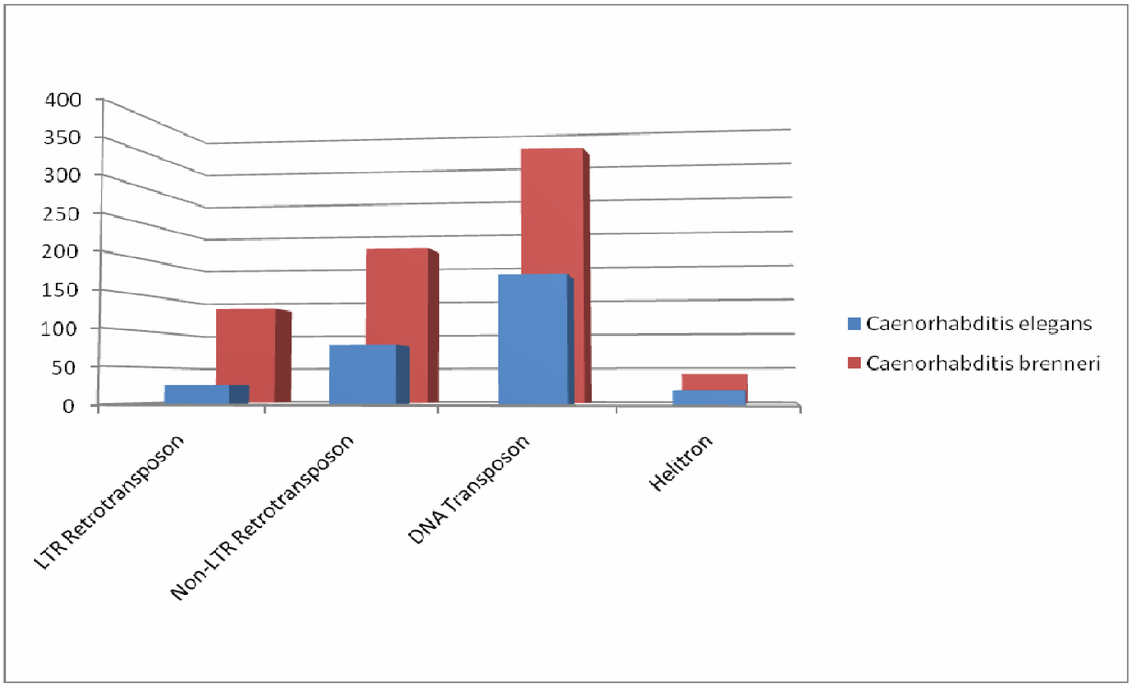


Figure 2.26: Comparison of number of repeat families classified by REPCLASS for *C. elegans* to *C. brenneri*

One of the most useful applications of REPCLASS is that it rapidly an accurate profile of TE diversity for a given genome defined by the relative amount and contribution of four major sub-classes of eukaryotic TEs: LTR retrotransposons, Non-LTR retrotransposons, DNA transposons and Helitrons. Figure 2.26 displays the comparison of the number of repeat families classified by REPCLASS for *C. elegans* and *C. brenneri* for each of these four categories. These classification numbers are for consensus libraries obtained using RS and after removing tandem repeats and low complexity repeats.

It is apparent from figure 2.26, that there is a significantly larger number of repeat families in all the 4 subclasses for *C. brenneri* than for *C. elegans*. Overall there was a 3.7 fold increase in the number of TE families classified by REPCLASS (139 families vs. 508 families). The increase affects all TE subclasses, ranging from a 2.28 fold increase in the number of Helitron families to a 5.19 fold increase in the number of LTR retrotransposon families. These data are indicative of an overall increase of TE diversity in *C. brenneri*. This difference is also apparent prior to REPCLASS classification at the level of the raw RS output, both before and after the filtering steps (428 consensus in *C. elegans* vs. 3392 in *C. brenneri*, after filtering). These results were unepxected because the total genome size of the two nematodes is not dramatically different (give total size of input sequence for both species), around 100 Mbp for *C. elegans* and close to 170 Mbp for *C. brenneri*. The average copynumber per repeat families is 115 for *C. elegans* and just 69 for *C. brenneri* (detailed view, table 2.4). Hence, we can conclude that an increase in TE diversity is not accompanied by an increase in total number of TEs in the *C. brenneri* genome.

We wondered whether the vast difference in the number of repeat families detected by RS and classified by RC in the two species could be an artifact introduced by the disparity in quality of genome sequences used as input. For *C. brenneri*, we used as input a 9.5-X whole genome shotgun assembly (WSGA) [49], while for *C. elegans* we input a high quality sequenced genome fully assembled into chromosomes. Intuitively RS might be expected to retrieve more fragmented repeats in a low-coverage whole genome shotgun assembly than in a high-quality genome sequence assembled into few chromosomes (6 chromosomes for *C. elegans*). Increased fragmentation would result in an inflated number of families accompanied by an overall shortening of consensus sequences. To test this possibility, we compared the average length of the consensus sequences classified by RC for each TE subclasses in C. brenneri and C. elegans.

Based on this data, it is clear that differential fragmentation and the discrepancy in quality of genome sequences cannot fully account for the increased number of TE families found in C. brenneri. Thus, we conclude that the results reflects a real biological difference in the TE landscape of these two nematode species [50].

Table 2.4: Average length comparison of *C. elegans* and *C. brenneri* repeat families

Subclass	C. elegans	C. brenneri
DNA	649	558
LTR	1403	1073
NON-LTR	707	478
HELITRON	891	1038

Regardless of the amount of families recovered, we observe that the relative proportion of each of the TE subclasses is very similar in the two genomes (figure 2.27) Thus, there is a strong consevration of the overall TE profile in these two distant nematode species despite substantial differences in the number of TE copies and families. These data support the hypothesis that the TE profile of a species is not the result of random and stochastic bursts of activity of different type of TEs, but rather that some forces are acting to shape and maintain TE composition for long period of evolutionary time [50].

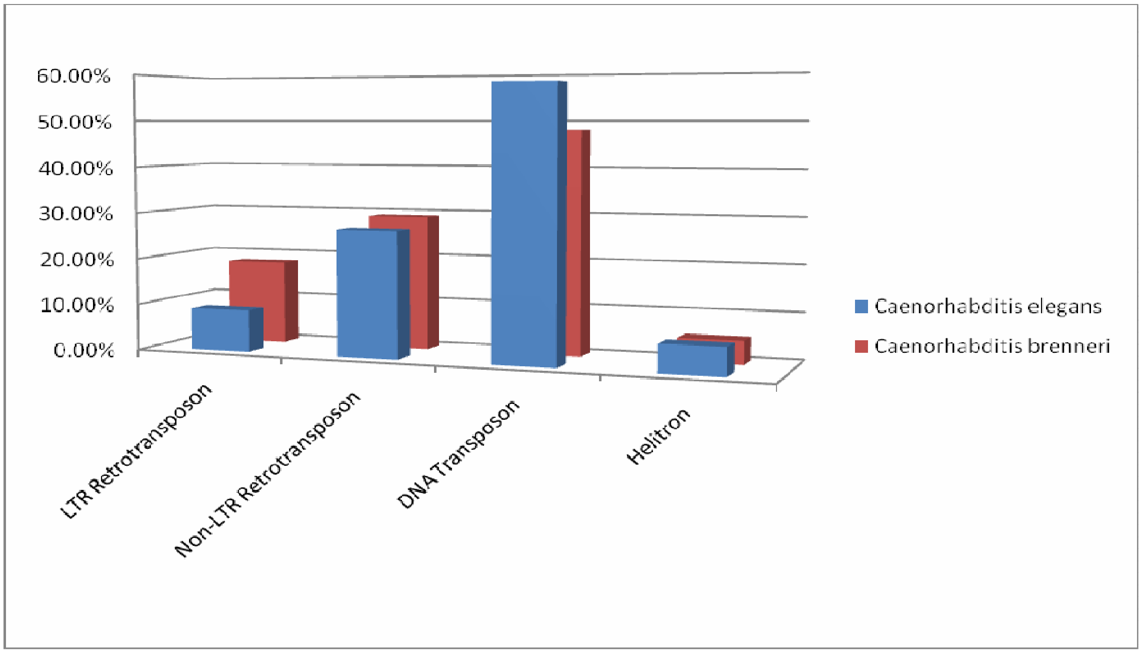


Figure 2.27: Comparison of *C. elegans* and *C. brenneri* TE profiles

2.5.2.3 Drosophila pseudoobscura

RepeatScout generated a consensus library of 4363 de novo repeat families. TRF and nseg discarded 75 sequences. After removing all those consensus sequences that were less then equal to or less than 100 bp long (figure 2.28) and filtering sequences with less than or equal to 10 copies (figure 2.29) in the genome, we were left with 1673 consensus sequences. REPCLASS classified 877 repeat families, more than 50% of the sequences identified by RepeatScout.

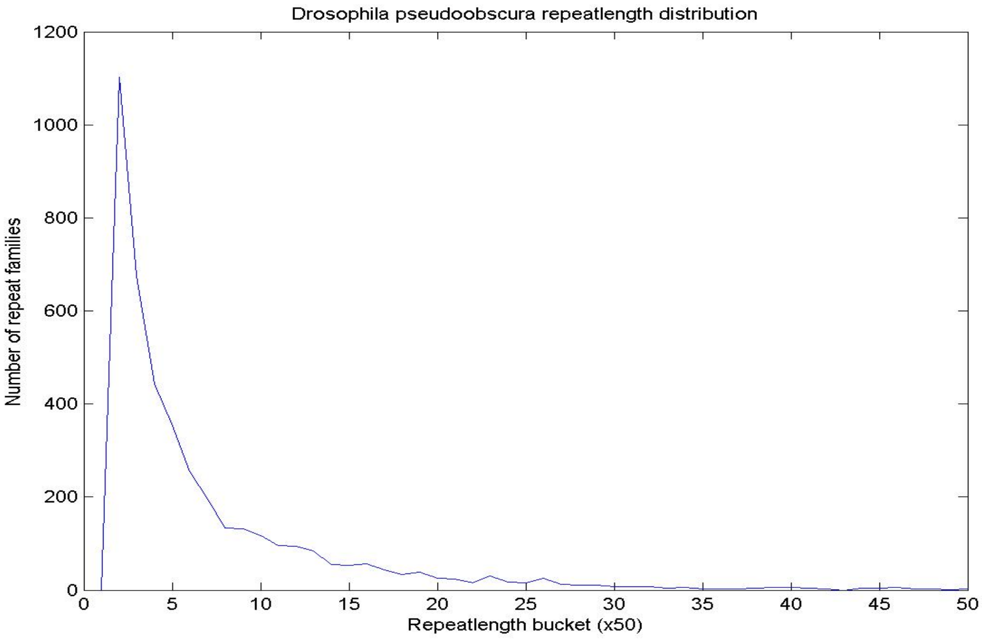


Figure 2.28: *D. pseudoobscura* repeatlength distribution

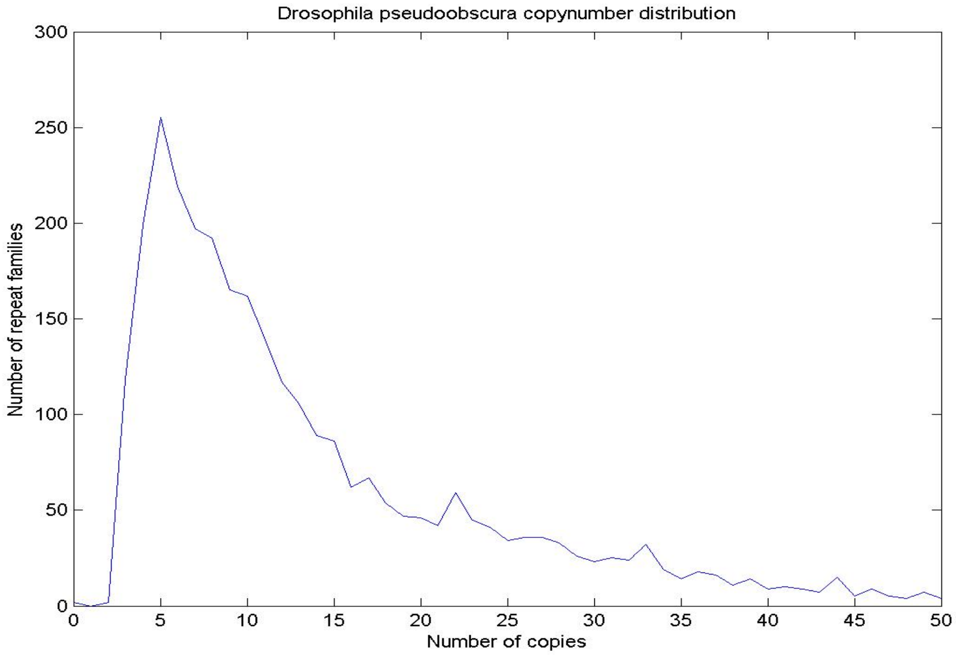


Figure 2.29: *D. pseudoobscura* copynumber distribution

2.5.2.4 *D. melanogaster* and *D. pseudoobscura* TE Profile Comparison

As earlier, here, we compared the number of repeat families classified by REPCLASS and the TE profile for the *D. melanogaster* and *D. pseudoobscura* genomes. Similar to the *C. elegans* and *C. brenneri* comparison graph, in the figure 2.28, we see that there is a significant increase in the number of repeat families classified for *D. pseudoobscura* than for *D. melanogaster*. Overall there was a two fold increase in the number of TE families classified by REPCLASS (815 families vs. 1671 families). The genome size of both these species is nearly the same, ~120 mbp for *D. melanogaster* and ~140 mbp for *D. pseudoobscura*.

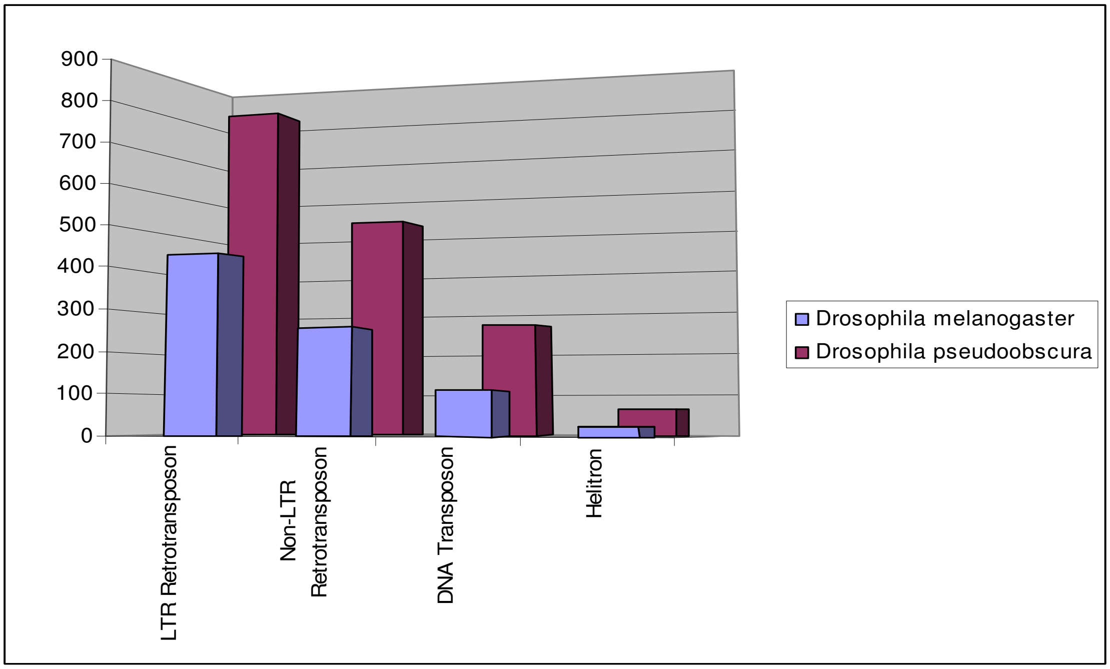


Figure 2.30: Comparison of number of repeat families classified by REPCLASS for *D. melanogaster* to *D. pseudoobscura*

To further understand this difference, we compared the average length of the consensus sequences classified by REPCLASS for each TE subclasses in *D. melanogaster* and *D. pseudoobscura*. We observe that, average length for LTRs and non-LTRs in *D. pseudoobscura* is nearly half of that for *D. melanogaster* (detailed view, see table 2.5). There are clear indications of fragmentation in the consensus sequences obtained by RepeatScout for the *D. pseudoobscura* genome [50].

Table 2.5: Average length comparison of *D. melanogaster* and *D. pseudoobscura* repeat families

Subclass	D. melanogaster	D. pseudoobscura
DNA	508	330
LTR	1411	766
NON-LTR	906	519
HELITRON	433	444

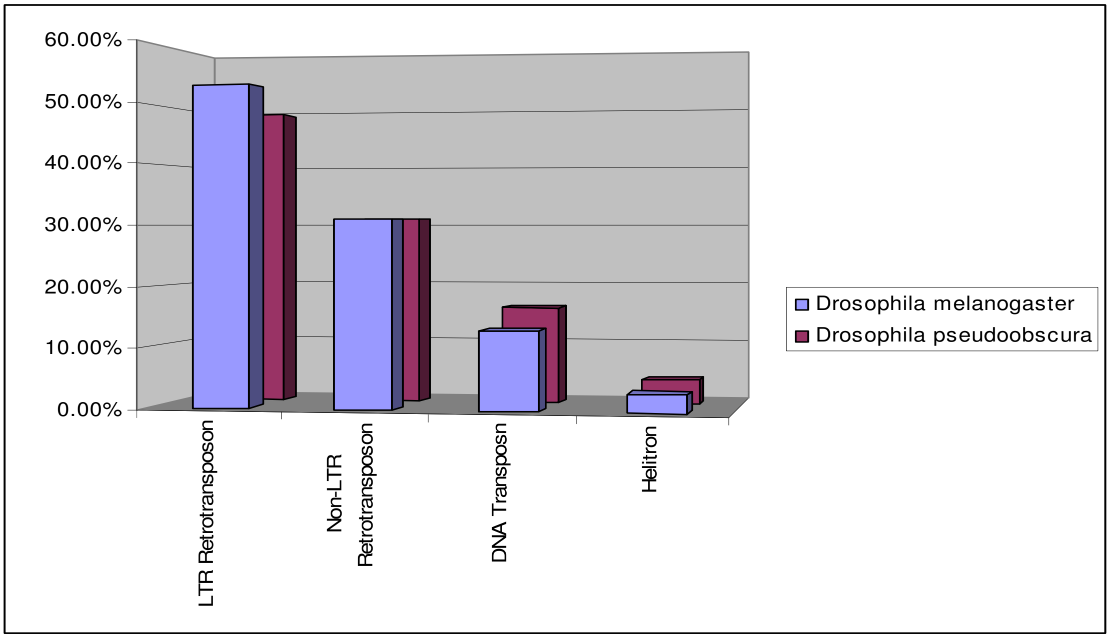


Figure 2.31: Comparison of D.melanogaster and D.pseudoobscura TE profiles

However, the overall TE profile of both the genomes is very similar, as confirmed by the comparative graph in figure 2.31. Hence, this data also supports the hypothesis that the TE profile of a species is not the result of random and stochastic bursts of activity of different type of TEs, but rather that some forces are acting to shape and maintain TE composition for long period of evolutionary time.

2.5.2.5 Classification of Repeats in Fungi

Increasingly in considering the genomes for TE classification, we have selected those genomes that serve less towards verification of already documented information hence the REPCLASS accuracy and more towards new and interesting discoveries. First we classified well-documented *C.elegans* and *D.melanogaster* libraries from Repbase. This was purely a control/verification measure. Then we moved on to de novo classification of *C.elegans* and *D.melanogaster* genome using consensus libraries obtained from RepeatScout. This was part a control measure and part new biology. We verified the RepeatScout/REPCLASS combined performance by showing that for both the genomes we could classify more than half the repeat families annotated in Repbase. We also made some new discoveries by classifying new TE families in genomes that have been extensively studied for decades. Following that, we classified the *C.brenneri* and *D.pseudoobscura* genomes. TEs in both these genomes have never been annotated before. We also selected these genomes because they are not too distantly related to genomes that we had already successfully classified. Hence, this step also helped as in validating the REPCLASS performance. Finally, continuing the same approach in this final result sub-section, we not only are considering genomes that have never been annotated in terms of TEs and but also are not related to any of the genomes that we have already considered.

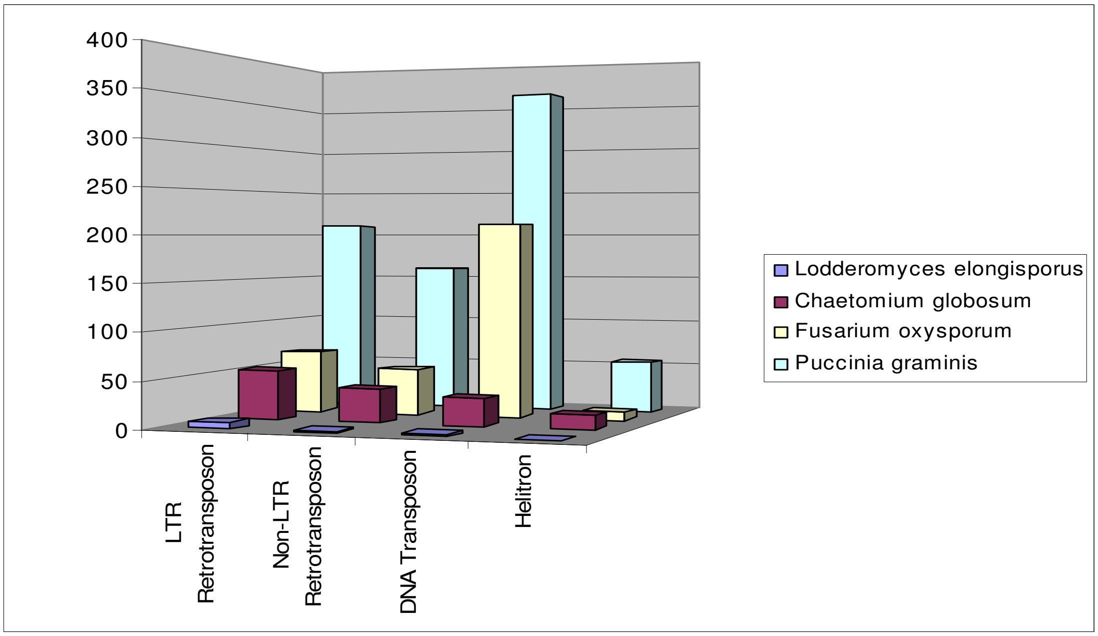


Figure 2.32: Comparing number of repeat families classified by REPCLASS among fungi

Here, we consider four fungi genomes namely *Lodderomyces elongisporus*, *Chaetomium globosum*, *Fusarium oxysporum* and *Puccinia graminis*. Fungi genomes we have considered have gradually increasing genome sizes (measured in terms of base pairs) and thus might be expected to containing proportionally increasing amount of TEs. Application of the RepeatScout/REPCLASS suite should allow us to test directly this hypothesis and also to determine whether such an increase is associated with an overall increase of TE diversity (i.e. increase in the number of TE families) or with an increase in copy numbers of only a few families. Finally, REPCLASS will provide an overview of the TE profile of each species and allow us to determine whether all subclasses are equally involved in changes in repeat content associated with genome size variation among fungi. *L. elongisporus*, a member of the *saccharomycetale* class of yeasts, has the smallest genome size (16 Mbp). *C. globosum* (class *Sordariomycetes*) is next at 35 Mbp followed by *F. oxysporum* (class *Sordariomycetes*) at 61 Mbp and finally *P. graminis* (class *Uredinales*) the biggest at 89 Mbp. One important fact to note is that all the fungi considered have evolved separately for more than 1000 million years [48]. In the figure 2.32 that shows graphs comparing the number of repeat families classified by REPCLASS for each of the fungi genomes, we can observe that as the size of the genome increases from *L. elongisporus* to *P. graminis* the number of repeat families classified also increases. This is in line with the earlier mentioned fact that size of the genomes co-relates to the number of TEs in the genomes.

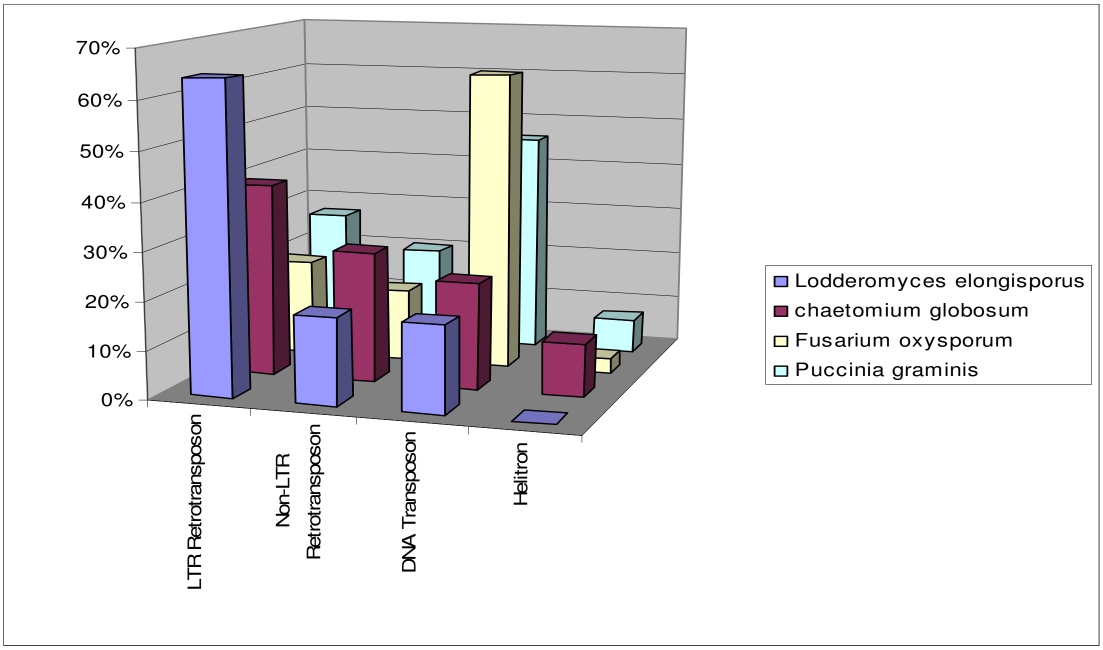


Figure 2.33: TE profile comparison for fungi

Figure 2.33 shows a comparative view of the TE profile of fungi genomes. We can observe that as the size of the genome increases the percentage of ltr retrotransposons in the genome decreases whereas the percentage of DNA transposons in the genome increases.

2.6 Conclusion

- REPCLASS can be used to quickly and accurately classify TEs for any eukaryotic species.
- REPCLASS is a tool that can discover new TE families and super-families, even in genomes that have been extensively studied.
- REPCLASS is able to capture accurately the very different TE profiles of *Caenorhabditis*, *Drosophila* and humans. Hence, we can conclude that RepeatScout/REPCLASS suite would work well for any eukaryotic species.
- Analysis of distantly related species of worm (*C. elegans* and *C. brenneri*) and flies (*D. melanogaster* and *D. pseudoobscura*) reveals that their TE profiles are conserved over a large evolutionary distance (more than 30 million years for both comparisons), despite the lack of nucleotide sequence conservation in the repeats identified.
- RepeatScout/REPCLASS comparative analysis of very divergent (more than 1000 million years apart) fungal genomes having varied genome sizes reveal the following:
  - TE diversity and number of TEs increase with genome size

    - Across wider evolutionary distance, we observe more dramatic changes in the TE composition of species

- After running all these experiments, we are now in a good position to analyze hundreds of genomes available and thousands that will be released soon.

CHAPTER 3

GRID COMPUTING SOLUTION FOR HIGH ENERGY PHYSICS

3.1 Background

3.1.1 High Energy Physics

High Energy Physics (HEP) also known as particle physics is the branch of physics that studies the fundamental constituents of matter e.g. sub-atomic particles. These fundamental particles are not observed under normal circumstances in nature. They are created and detected during highly accelerated collisions of other larger particles. These high-energy collisions take place in huge particle accelerators. The biggest particle accelerator in the world today is the Large Hadron Collider (LHC) [28].

3.1.2 LHC

LHC is located near Geneva, where it spans the border between Switzerland and France about 100 m underground. The European Organization for Nuclear Research (CERN laboratory), a European joint venture of 20 member states build it. LHC is an underground ring of 27km in circumference. Two beams of subatomic particles called 'hadrons' – either protons or lead ions – will travel in opposite directions inside the circular accelerator, gaining energy with every lap and finally colliding head-on at very high energy. Physicists from around the world will analyze the particles created in the collisions using special detectors in a number of experiments.

There are six experiments at the LHC. All of the experiments are run by collaborations of scientists from all over the world. Each experiment is distinct, characterized by its unique particle detector. The two large experiments, ATLAS and CMS [29], are based on general-purpose detectors to analyze the myriad of particles produced by the collisions in the accelerator. They are designed to investigate the largest range of physics possible. Having two independently designed detectors is vital for cross-confirmation of any new discoveries made. Two medium-size experiments, ALICE [30] and LHCb [31], have specialized detectors for analyzing the LHC collisions in relation to specific phenomena. Two experiments, TOTEM [32] and LHCf [33], are much smaller. They are designed to focus on 'forward particles' (protons or heavy ions). These are particles that just brush past each other as the beams collide, rather than meeting head-on. Here, we would talk about the ATLAS in detail.

3.1.3 ATLAS

ATLAS is one of the largest collaborative efforts ever attempted in physics. Currently, there are about 2100 physicists participating from more than 167 universities and laboratories in 37 countries. Starting later this year, the ATLAS experiment will start generating data, searching for new discoveries in the head-on collisions of protons traveling at extraordinarily high speed. Discovering new fundamental particles and fields and analyzing their properties is possible through statistical analysis of the massive amounts of data gathered by the ATLAS detector inside the LHC and its detailed comparison with compute-intensive theoretical simulations. Figure 3.1 shows a detailed computer-generated image of the ATLAS detector.

The ATLAS detector will collect an enormous amount of data, about 100 kilobytes every 12 nanoseconds. To help digest this data the ATLAS experiment operates the trigger system which selects 100 interesting events per second out of the almost 100 million collected, the data acquisition system channels the data from the detector to the storage and the computing system analyzes more than 1000 million events recorded every year.

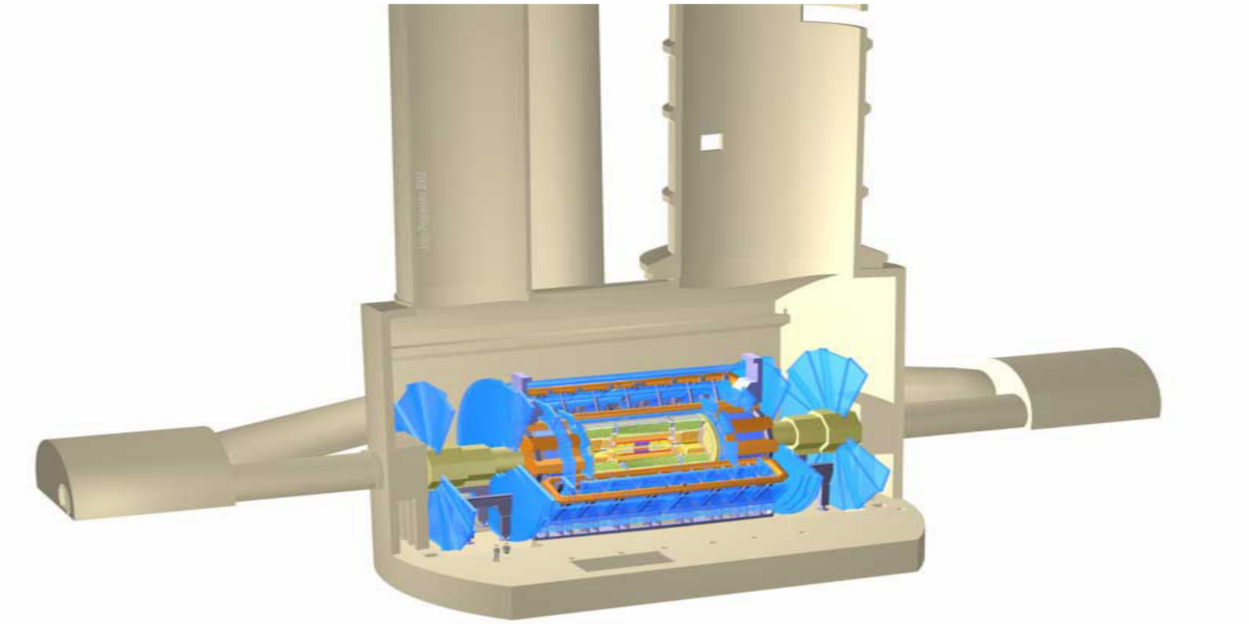


Figure 3.1: A detailed computer-generated image showing a part of LHC and the ATLAS detector [7].

Traditionally, for an experiment such as ATLAS that has gigantic computing and storage requirements a centralized model may be selected. However, in the case of ATLAS, a globally distributed model for data storage and analysis is preferred. A computing and data grid, a model that would provide the ability to perform high throughput computing and distributed data storage by taking advantage of many networked computers and storage devices around the world. One of the motivating factors for adopting a distributed model was the huge cost of maintaining and upgrading the necessary resources for such a computing challenge. These costs could be better managed when contributed to by participating universities and research institutes by funding local resources while still helping achieve the global goal. In addition, there are other advantages to having a computing and data grid; having distributed computing resources leads to better load balancing, efficient use of available resources, avoids bottlenecks and leads to data security by replicating the data at multiple independent geographical locations.

The data generated by the ATLAS experiment will be distributed around the globe, according to a four-tiered model. A primary backup will be recorded on tape at CERN, the highest (tier-0) centre. After initial processing, this data will be distributed to a number of Tier-1 centers, large computer centers with sufficient storage capacity and huge computing power geographically spread across the world. The Tier-1 centers will make data available to Tier-2 centers, each consisting of one or several collaborating computing facilities, which can store sufficient data and provide adequate computing power for specific analysis tasks. Individual scientists will access these facilities through Tier-3 computing resources, which can consist of local clusters in a university department or even individual PCs, which may be allocated for ATLAS data storage and analysis on a regular basis.

The Brookhaven National Laboratory (BNL) in New York, an ATLAS Tier-1 centre and all the associated Tier-2 and Tier-3 centers located across the US are known as US ATLAS. To meet the challenging ATLAS computing requirements, US ATLAS developed a production (running simulations, processing data to make it analyzable) and distributed analysis system called Panda. Panda runs independent of any grid middleware. It generally runs on the Open Science Grid (OSG) [34], a consortium of software and resource provider universities and research institutes across the US. Panda is also capable of running on the LHC Computing Grid (LCG) [35], TeraGrid [36] and other similar grid initiatives.

3.1.4 PANDA

Panda submits jobs to computing nodes all across US. These nodes are part of different clusters (sites) and sometimes part of different grids. These clusters run different batch systems like Condor [37], Portable Batch System (PBS), and other batch systems. Batch system is software that allocates jobs or tasks to be executed to all the computers in a cluster. In order to be a successful grid computing middleware, it is important that Panda is able to communicate with all the batch systems effectively using a common protocol. Panda achieves this by using Condor-G [38] and the Globus Toolkit [39], with GRAM [40] being the common communication protocol. Condor-G is the grid ready version of Condor and is able to communicate with Globus using

GRAM. Condor is a specialized workload management system for compute-intensive jobs whereas Globus is an open source software toolkit used for building grids. Every site in the grid has a Globus process running on the gatekeeper or the head node. Gatekeeper or head node in a cluster is a machine that is enabled for outside access. Every time Panda submit a job to a site remotely, it is delivered to the site by Condor-G. Condor-G communicates this job definition to the Globus process running on the head node via GRAM. Globus in turn spawns a new job-manager process that converts this job definition to the format required by the native batch system. The native batch system then schedules and executes the job.

3.2 AutoPilot

Panda was designed for handling jobs for the ATLAS experiment and hence some of the modules had experiment specific content. In September 2006, a new effort began in collaboration with the OSG to generalize Panda into a generic high-level workload manager usable by anyone in the OSG or the wider grid community. An important part of this new effort is AutoPilot [41]. AutoPilot is a simple and generic implementation of Panda pilot and pilot-scheduler for use in more varied environments than currently in use within Panda. Panda pilot is a lightweight execution environment used to prepare the computing resources for job execution, request the actual payload (a production or user analysis job) from the Panda server, execute it, and clean up when the job is done. These pilots are broadcasted from the pilot-scheduler to all the grid sites.

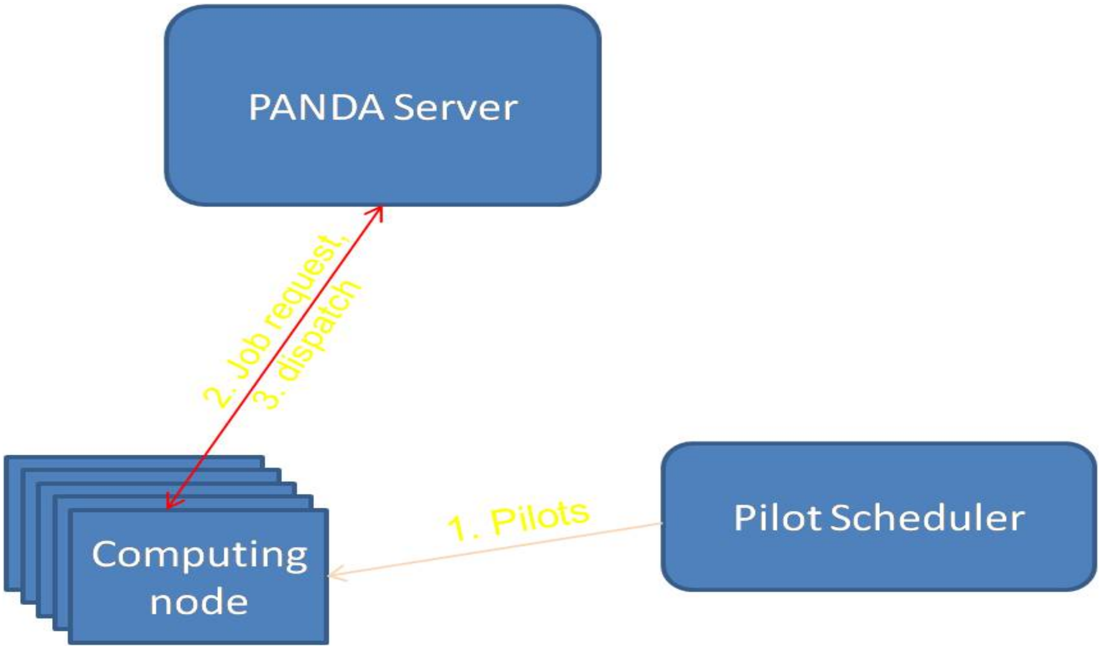


Figure 3.2: Simplistic representation of PANDA workflow

AutoPilot's pilot-scheduler implementation although generic continues the same Panda methodology. A centrally operated pilot-scheduler sending pilots to all the participating sites across the grid. However, this approach has encountered scaling limitations in sending pilots to a site. With the increase in the number of production and user jobs being executed, more pilots are submitted to service and prepare these jobs. This in turn has lead to very heavy GRAM traffic. In addition, each running pilot job on the site requires a job-manager process on the head node of the site. As a result, more number of pilot jobs leads to large memory consumption in addition to the excessive computing power required on the head node.

To solve the above-mentioned problem we have to develop a process that submits pilots in a manner that reduces or, better yet, circumvents GRAM communication. One implement able solution is to delegate the pilot submission activity to one of the nodes within the site i.e. local submission of pilots within the cluster. This node will disseminate pilots locally to other worker nodes generating no GRAM traffic. This way, the excess computing and data storage demands on the head node will be eliminated. Pilot submission by this approach would require surrendering the job control over to the local sites, rendering it difficult for others to access the pilot output and the generated log files. This problem is solved by allowing access to the pilot output and log files through an external link such as HTTPS.

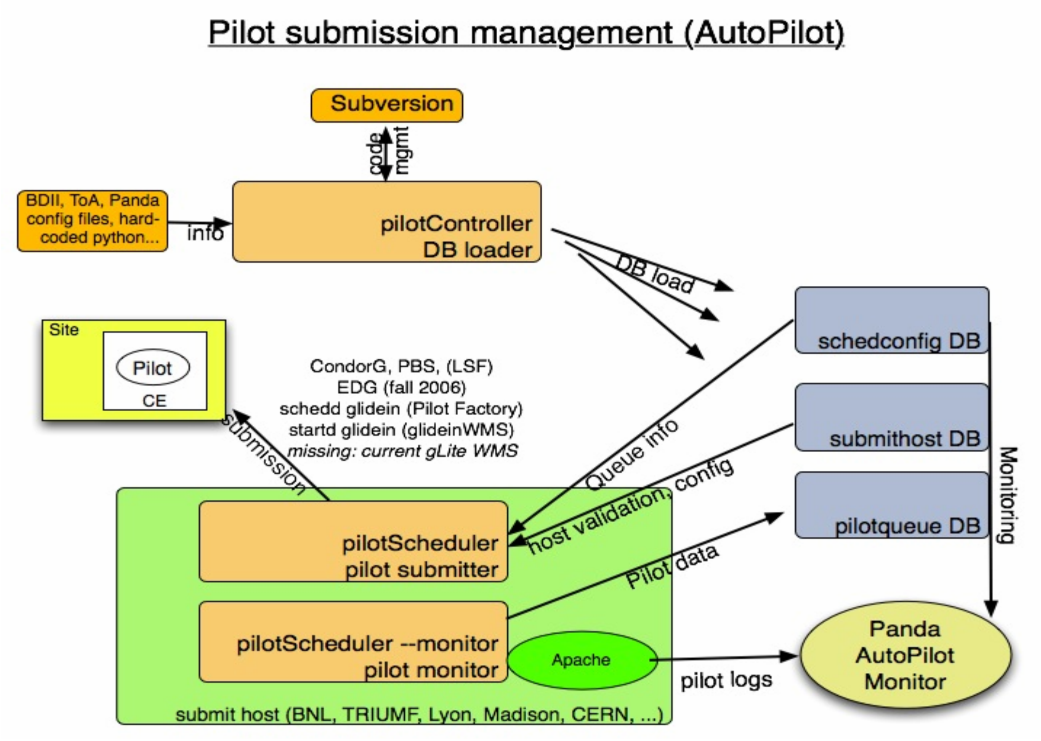


Figure 3.3: Detailed AutoPilot workflow.

A move from remote pilot submission to local pilot submission mechanism would not be a problem for sites running the Condor batch system. AutoPilot currently submits pilots using Condor-G commands, and these would work fine, even with Condor. However, it would not work for sites running any other batch system. Hence, the goal of this work is to enhance AutoPilot so that it is able submit pilots locally to sites running PBS.

3.3 PBS-Capable AutoPilot

PBS is an extensible batch queuing and job management system for the UNIX operating system. It is portable, very flexible and provides a graphical user interface. PBS consists of four major components. These are PBS commands, job Server, job executor, and job Scheduler. PBS commands are used to submit, monitor, modify, and delete jobs. These are classified into three categories, user commands (qsub, qstat, etc), operator commands (qenable, qdisable, etc) and administrator commands (qmgr, pbsnodes, etc). Operator and administrator commands require different access privileges than user commands.

PBS user commands qsub and qstat are the most important and most used. Qsub command submits a job. Job script to be executed is passed as an argument. It returns the job id, if the job is successfully submitted. It also has several user-options like '–q' to specify the destination queue you want the job to be submitted or '-I' which used if you want the job to run interactively and many others. The command 'qstat' gets the status of a running batch job. Job id of the job being enquired for is passed as an argument. Job has to be in the running state for this command to be successful.

In the AutoPilot architecture, pilotScheduler is the central process that both submits and monitors pilots. It interacts with a number of tables that store information about the sites known to AutoPilot, different scripts to be executed in order to submit pilots on different batch systems and other similar data. The pilotController process populates most of these tables. The pilotScheduler also writes to the database but that is only when it is monitoring pilots. It collects the monitoring data using Condor-G commands and keeps the database updated. pilotScheduler also writes out the pilot outputs and log files in a designated area on a local machine so that it can be accessed by anyone interested. Figure is a detailed look at the AutoPilot architecture.

To make AutoPilot capable of submitting pilots locally to sites running PBS with minimal changes we worked with the existing pilot scheduler framework provided by AutoPilot. Since, in addition to the submission capabilities AutoPilot architecture also provides for monitoring the submitted pilots, we had to make sure that not only the pilots were successfully submitted locally but also they were properly monitored and the monitoring data collected, was correctly communicated to the database so that it can be displayed on the central Panda AutoPilot Monitor [42].

To enable local PBS job submissions we introduce the PBS job submission command ‘qsub’ to AutoPilot and since, most sites have a dedicated queue for ATLAS jobs, we use the ‘–q’ option to specify the destination queue. To monitor the state of the job ‘qstat’ PBS command is used. The nature of the job status data communicated by PBS is different than Condor-G. Hence, we have to parse the PBS job status information separately and had to convert it to the format understood by the AutoPilot database. Once the job status information is entered in the database, it is picked by the AutoPilot monitoring mechanism and displayed on the Panda AutoPilot Monitor.

REFERENCES

[1] Cluster Computing Info Centre, http://www.buyya.com/cluster

[2] Grid Computing Info Centre, http://www.gridcomputing.com

[3] Ranganathan N, “REPCLASS: Cluster and Grid Enabled Automatic Classification of Transposable Elements Identified de novo in Genome Sequences”, Master's Theses, University of Texas at Arlington, December 2006.

[4] J. Finnegan. Eukaryotic transposable elements and genome evolution. *Trends in Genetics,* vol. 5, 103-107, 1989.

[5] Eukaryotes, http://www.tolweb.org/Eukaryotes

[6] Panda Twiki at CERN, https://twiki.cern.ch/twiki/bin/view/Atlas/Panda

[7] ATLAS experiment website, www.atlasexperiment.org

[8] PBS website, http://www.nas.nasa.gov/Software/PBS/

[9] Cells and DNA, http://ghr.nlm.nih.gov/handbook/basics

[10] DOEgenomes.org genomics primer, http://www.ornl.gov/sci/techresources/Human_Genome/publicat/primer2001/1.shtml

[11] Kapitonov VV, Jurka J. (2003) Molecular paleontology of transposable elements in the Drosophila melanogaster genome. Proc. Natl. Acad. Sci., 100: 6569-6574.

[12] Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, et al. (2001) Initial sequencing and analysis of the human genome. Nature 409: 860-921.

[13] M. G. Kidwell and D. R. Lisch. Perspective: Transposable Elements, Parasitic DNA, and Genome Evolution. Evolution, vol. 55(1), 1-24, 2001.

[14] N. J. Bowen and I. K. Jordan. Transposable Elements and evolution of eukaryotic complexity. Mol. Biol., vol. 4(3), 65-76, 2002

[15] C. Feschotte, N. Jiang, S. R. Wessler. Plant Transposable Elements: Where Genetics Meets Genomics. Nature Reviews Genetics, vol. 3, 329-341, 2002.

[16] P. L. Deininger, J. V. Moran, M. A. Batzer and H. H. Kazazian. Mobile elements and mammalian genome evolution. Curr. Opin. Genet. Dev., vol. 13, 651-658, 2003.

[17] Harshey RM and Bukhari AI (1981) A mechanism of DNA transposition. Proc. Natl. Acad. Sci., 78: 1090-1094.

[18] Bao Z and Eddy SR. (2002) Automated de novo Identification of Repeat Sequence Families in Sequenced Genomes. Genome Research, 12: 1269-1276.

[19] Price AL, Jones NC and Pevzner PA. (2005) De novo identification of repeat families in large genomes. Bioinformatics, 21: i351-i358.

[20] Edgar RC and Myers EW. (2005) PILER: identification and classification of genomic repeats. Bioinformatics 21(Suppl 1): i152-i158.

[21] Li R, Ye J, Li S, Wang J, Han Y, et al. (2005) ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. PLoS Computational Biology 1(4): E43.

[22] Lipman DJ and Pearson WR. (1985) Rapid and Sensitive Protein Similarity Searches. Science, 227: 1435-1441.

[23] Benson G, (1999). Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Research, 27: 573-580.

[24] Wooton JC and Federhen S. (1993) Statistics of local complexity in amino acid sequences and sequence databases. Computers and Chemistry, 17: 149–163.

[25] Kapitonov VV, Jurka Jerzy. (2001) Rolling-circle transposons in eukaryotes. Proc. Natl. Acad. Sci., 98: 8714-8719.

[26] Feschotte C, Wessler SR. (2001) Treasures in the attic: Rolling circle transposons discovered in eukaryotic genomes. Proc. Natl. Acad. Sci., 98: 8923-8924.

[27] Emboss (The European Molecular Biology Open Software Suite), http://emboss.sourceforge.net

[28] LHC website, http://public.web.cern.ch/public/en/LHC/LHC-en.html

[29] The Compact Muon Solenoid Experiment, http://cms.cern.ch/

[30] A Large Ion Collider Experiment, http://aliceinfo.cern.ch/

[31] The Large Hadron Collider beauty experiment, http://lhcb.web.cern.ch/lhcb/

[32] TOTEM experiment homepage, http://totem.web.cern.ch/Totem/

[33] LHCf experiment homepage, http://hep.fi.infn.it/LHCf/index.html

[34] Open Science Grid homepage, http://www.opensciencegrid.org/

[35] LHC Computing Grid Project, http://lcg.web.cern.ch/LCG/

[36] TeraGrid homepage, http://www.teragrid.org/index.php

[37] Condor project homepage, http://www.cs.wisc.edu/condor/

[38] Condor-G: A Computation Management Agent for Multi-Institutional Grids. J. Frey, T. Tannenbaum, M. Livny, I. Foster, S. Tuecke. *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*, IEEE Press, August 2001.

[39] Globus Toolkit Version 4: Software for Service-Oriented Systems. I. Foster. *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS 3779, pp 2-13, 2006.

[40] GRAM webpage http://dev.globus.org/wiki/GRAM

[41] AutoPilot Twiki at BNL http://www.usatlas.bnl.gov/twiki/bin/view/AtlasSoftware/AutoPilot

[42] Panda AutoPilot Monitor
http://gridui02.usatlas.bnl.gov:25880/server/pandamon/query?tp=main

[43] Thomas Wicker, et al. (2008). A universal classification of eukaryotic transposable elements implemented in Repbase, Nature Reviews Genetics 9, 414 – 414.

[44] GOLD: Genomes OnLine Database, http://www.genomesonline.org/gold.cgi

[45] Saha, et al. (2008). Empirical comparison of *ab initio* repeat finding programs, Nucleic Acids Research, Vol. 36, No. 7 2284-2294

[46] Gish WR. Washington University BLAST, http://blast.wustl.edu/

[47] Jurka, J., Kapitonov, V.V., Pavlicek, A., Klonowski, P., Kohany, O., Walichiewicz, J. (2005) Repbase Update, a database of eukaryotic repetitive elements. Cytogentic and Genome Research 110:462-467

[48] S. Blair Hedges. (2002). The origin and evolution of model organisms**,** Nature Reviews Genetics 3, 838 – 849.

[49] Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J. F., Dougherty, B. A., Merrick, J. M., et al. (1995) Science 269, 496–512.

[50] Feschotte C (personal communication), 2008.

BIOGRAPHICAL INFORMATION

Umeshkumar Keswani joined the University of Texas at Arlington in the fall of 2006. He received his Bachelor's degree in Computer Engineering from Thadomal Shahani Engineering College affiliated to the Mumbai University, India. His research interests are computer architecture and high performance computing. He received his M.S in Computer Science and Engineering in 2008.